The BalaBit Shell Control Box 3 LTS Administrator Guide

Publication date February 14, 2013

Abstract
This document is the primary manual of the BalaBit Shell Control Box 3 LTS.

Revision History
For a list of documentation changes, see *Section 6, About this document (p. xv)*.

Copyright © 1996-2013 BalaBit IT Security Ltd.

This guide is published under the Creative Commons Attribution-Noncommercial-No Derivative Works (by-nc-nd) 3.0 license. See *Appendix 7, Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd) License (p. 263)* for details. The latest version is always available at *http://www.balabit.com/support/documentation*.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (*http://www.openssl.org/*). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)

This documentation and the product it describes are considered protected by copyright according to the applicable laws.

The Zorp™ name and the Zorp™ logo are registered trademarks of BalaBit.

The BalaBit Shell Control Box™ name and the BalaBit Shell Control Box™ logo are registered trademarks of BalaBit.

The syslog-ng™ name and the syslog-ng™ logo are registered trademarks of BalaBit.

The BalaBit™ name and the BalaBit™ logo are registered trademarks of BalaBit.

Linux™ is a registered trademark of Linus Torvalds.

Debian™ is a registered trademark of Software in the Public Interest Inc.

Windows™ 95, 98, ME, 2000, XP, Server 2003, Vista, Server 2008 and 7 are registered trademarks of Microsoft Corporation.

Linux™ is a registered trademark of Linus Torvalds.

Debian™ is a registered trademark of Software in the Public Interest Inc.

Windows™ 95, 98, ME, 2000, XP, Server 2003, Vista, and Server 2008 are registered trademarks of Microsoft Corporation.

MySQL™ is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Oracle™, JD Edwards™, PeopleSoft™, and Siebel™ are registered trademarks of Oracle Corporation and/or its affiliates.

Sun™, Sun Microsystems™, the Sun logo, Sun Fire 4140™, Sun Fire 2100™, Sun Fire 2200™, Sun Fire 4540™, and Sun StorageTek™ are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the U.S. and other countries.

VMware™, VMware ESX™ and VMware View™ are trademarks or registered trademarks of VMware, Inc. and/or its affiliates.

Citrix®, ICA® and XenApp™ are trademarks or registered trademarks of Citrix Systems, Inc.

All other product names mentioned herein are the trademarks of their respective owners.

Some rights reserved.

DISCLAIMER

BalaBit is not responsible for any third-party Web sites mentioned in this document. BalaBit does not endorse and is not responsible or liable for any content, advertising, products, or other material on or available from such sites or resources. BalaBit will not be responsible or liable for any damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through any such sites or resources.

Table of Contents

Preface ..... **xii**
1. Summary of contents xii
2. Target audience and prerequisites xiii
3. Products covered in this guide xiii
4. Typographical conventions xiii
5. Contact and support information xiv
5.1. Sales contact xiv
5.2. Support contact xiv
5.3. Training xv
6. About this document xv
6.1. Feedback xv
1. Introduction **1**
1.1. What SCB is 1
1.2. What SCB is not 2
1.3. Why is SCB needed? 2
1.4. Who uses SCB? 2
1.5. Public references for BalaBit Shell Control Box: 3
2. The concepts of SCB **4**
2.1. The philosophy of SCB 4
2.2. Supported protocols and client applications 5
2.3. Modes of operation 6
2.3.1. Bridge mode 6
2.3.2. Router mode 8
2.3.3. Single-interface router mode 9
2.3.4. Bastion mode 10
2.3.5. Nontransparent mode 11
2.4. Connecting to a server through SCB 12
2.4.1. Connecting to a server through SCB using SSH 12
2.4.2. Connecting to a server through SCB using RDP 227
2.5. SSH hostkeys 16
2.6. Authenticating clients using public-key authentication in SSH 16
2.7. The gateway authentication process 16
2.8. 4-eyes authorization 17
2.9. Network interfaces 18
2.10. High Availability support in SCB 19
2.11. Firmware in SCB 20
2.11.1. Firmwares and high availability 20
2.12. Versions and releases of SCB 20
2.13. Accessing and configuring SCB 21
2.14. Licenses 21
3. The Welcome Wizard and the first login **23**
3.1. The initial connection to SCB 23
3.1.1. Creating an alias IP address (Microsoft Windows) 24
3.1.2. Creating an alias IP address (Linux) 28
3.1.3. Modifying the IP address of SCB 28

3.2. Configuring SCB with the Welcome Wizard 29
3.3. Logging in to SCB and configuring the first connection 37
4. Configuring and managing SCB 41
4.1. Supported web browsers and operating systems 41
4.2. The structure of the web interface 41
4.2.1. Elements of the main workspace 44
4.2.2. Multiple web users and locking 45
4.3. Basic settings 45
4.3.1. Network settings 46
4.3.2. Date and time configuration 50
4.3.3. System logging, SNMP and e-mail alerts 52
4.3.4. Configuring system monitoring on SCB 57
4.3.5. Data and configuration archiving and backups 62
4.4. User management and access control 75
4.4.1. Managing SCB users locally 75
4.4.2. Setting password policies for local users 76
4.4.3. Managing local usergroups 78
4.4.4. Managing SCB users from an LDAP database 79
4.4.5. Authenticating users to a RADIUS server 82
4.4.6. Managing user rights and usergroups 84
4.4.7. Listing and searching configuration changes 90
4.5. Managing SCB 91
4.5.1. Controlling SCB — restart, shutdown 91
4.5.2. Managing a high availability SCB cluster 93
4.5.3. Upgrading SCB 103
4.5.4. Accessing the SCB console 110
4.5.5. Sealed mode 113
4.5.6. Out-of-band management of SCB 114
4.5.7. Managing the certificates used on SCB 116
5. Configuring connections 120
5.1. General connection settings 120
5.1.1. Configuring connections 120
5.1.2. Modifying the destination address 123
5.1.3. Modifying the source address 125
5.1.4. Creating and editing channel policies 126
5.1.5. Configuring time policies 129
5.1.6. Creating and editing user lists 130
5.1.7. Authenticating users to an LDAP server 131
5.1.8. Audit policies 135
5.1.9. Verifying certificates with Certificate Authorities 142
5.1.10. Signing certificates on-the-fly 143
5.1.11. Forwarding traffic to an IDS or DLP system 144
5.1.12. Configuring cleanup for the SCB connection database 147
5.2. SSH-specific settings 148
5.2.1. Setting the SSH host keys and certificates of the connection 149
5.2.2. Supported SSH channel types 151
5.2.3. Authentication Policies 154
5.2.4. Server host keys and certificates 161

5.2.5. Creating and editing protocol-level SSH settings 163
5.3. RDP-specific settings 165
5.3.1. Supported RDP channel types 165
5.3.2. Creating and editing protocol-level RDP settings 166
5.3.3. Joining SCB into a domain 168
5.3.4. Using SSL-encrypted RDP connections 169
5.3.5. Verifying the certificate of the RDP server in encrypted connections 170
5.4. Telnet-specific settings 172
5.4.1. Creating and editing protocol-level Telnet settings 172
5.5. VNC-specific settings 172
5.5.1. Creating and editing protocol-level VNC settings 173
5.6. VMware View connections 173
6. Browsing log messages and SCB reports 175
6.1. Using the search interface 175
6.1.1. Displaying information about connections 176
6.2. Changelogs of SCB 178
6.3. Reports 179
6.4. The SCB connection database 181
6.4.1. Connection metadata 182
6.4.2. Creating predefined filters 185
6.5. Configuring custom reports 185
6.6. Monitoring the status of AP indexing services 188
6.7. Configuring full-text indexing of audit trails 189
7. Viewing session information and replaying audit trails 191
7.1. Installing the Audit Player application 191
7.2. Replaying audit trails 194
7.2.1. Downloading audit trails from SCB 194
7.2.2. Replaying a session with the Audit Player 195
7.2.3. Replaying SCP and SFTP sessions 197
7.3. Using AP 198
7.3.1. Finding specific audit trails 200
7.3.2. Using projects 202
7.3.3. Replaying and processing encrypted audit trails 202
7.3.4. Searching in graphical streams 204
7.3.5. Adding a new font to the OCR database 204
7.4. Troubleshooting the Audit Player 205
7.4.1. Logging with the Audit Player 205
7.4.2. Keys and certificates 206
7.4.3. Keyframe building errors 207
8. Advanced authentication and authorization techniques 208
8.1. Configuring usermapping policies 208
8.2. Configuring gateway authentication 210
8.2.1. Configuring outband gateway authentication 210
8.2.2. Performing outband gateway authentication on SCB 212
8.2.3. Performing inband gateway authentication in SSH connections 213
8.2.4. Performing inband gateway authentication in RDP connections 214
8.2.5. Troubleshooting gateway authentication 214
8.3. Configuring 4-eyes authorization 215

8.3.1. Configuring 4-eyes authorization 215
8.3.2. Performing 4-eyes authorization on SCB 218
9. Best practices and configuration examples 220
9.1. Configuring public-key authentication on SCB 220
9.1.1. Configuring public-key authentication using local keys 220
9.1.2. Configuring public-key authentication using an LDAP server and a fixed key 221
9.1.3. Configuring public-key authentication using an LDAP server and generated keys 222
9.2. Organizing connections in Bastion mode 223
9.2.1. Organizing connections based on port numbers 223
9.2.2. Organizing connections based on alias IP addresses 223
9.2.3. Accessing the SCB host in Bastion mode using SSH 224
9.3. Using nontransparent Bastion mode 224
9.4. Restoring SCB configuration and data 226
10. SCB scenarios 227
10.1. SSH usermapping and keymapping in AD with public key 227
11. Troubleshooting SCB 235
11.1. Network troubleshooting 235
11.2. Gathering data about system problems 236
11.3. Viewing logs on SCB 236
11.4. Changing log verbosity level of SCB 237
11.5. Collecting logs and system information for error reporting 238
11.6. Status history and statistics 240
11.6.1. Displaying custom connection statistics 241
11.7. Troubleshooting an SCB cluster 242
11.7.1. Recovering SCB if both nodes broke down 242
11.7.2. Recovering from a split brain situation 243
11.7.3. Replacing a HA node 245
11.8. Restoring configuration from a configuration backup 245
Appendix 1. About the Secure Shell protocol in a nutshell 247
1.1. The basic operation of SSH 247
1.2. Configuring encryption parameters 248
Appendix 2. Package contents inventory 249
Appendix 3. BalaBit Shell Control Box Hardware Installation Guide 250
3.1. Installing the SCB hardware 250
3.2. Installing two SCB units in HA mode 251
Appendix 4. BalaBit Shell Control Box VMware Installation Guide 253
4.1. Limitations of SCB under VMware 253
4.2. Installing SCB under VMware ESXi 253
Appendix 5. Configuring external devices 255
5.1. Configuring advanced routing on Linux 255
Appendix 6. BalaBit Shell Control Box End User License Agreement 257
6.1. SUBJECT OF THE LICENSE CONTRACT 257
6.2. DEFINITIONS 257
6.3. WORDS AND EXPRESSIONS 257
6.4. LICENSE GRANTS AND RESTRICTIONS 258
6.5. SUBSIDIARIES 259
6.6. INTELLECTUAL PROPERTY RIGHTS 259
6.7. TRADE MARKS 259

6.8. NEGLIGENT INFRINGEMENT 260
6.9. INTELLECTUAL PROPERTY INDEMNIFICATION 260
6.10. LICENSE FEE 260
6.11. WARRANTIES 260
6.12. DISCLAIMER OF WARRANTIES 261
6.13. LIMITATION OF LIABILITY 261
6.14. DURATION AND TERMINATION 261
6.15. AMENDMENTS 262
6.16. WAIVER 262
6.17. SEVERABILITY 262
6.18. NOTICES 262
6.19. MISCELLANEOUS 262
Appendix 7. Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd) License 263
Glossary 268
Index 272

List of Examples

4.1. Configuring NFS on the remote server 74
7.1. Using the `-C` switch 194

List of Procedures

2.4.1. Connecting to a server through SCB using SSH 12
2.4.2. Connecting to a server through SCB using RDP 227
2.7. The gateway authentication process 16
2.8. 4-eyes authorization 17
3.1.1. Creating an alias IP address (Microsoft Windows) 24
3.1.2. Creating an alias IP address (Linux) 28
3.1.3. Modifying the IP address of SCB 28
3.2. Configuring SCB with the Welcome Wizard 29
3.3. Logging in to SCB and configuring the first connection 37
4.3.1.1. Configuring the management interface 47
4.3.1.2. Routing management traffic to the management interface 49
4.3.2.1. Configuring a time (NTP) server 51
4.3.3.1. Configuring system logging 52
4.3.3.2. Configuring e-mail alerts 54
4.3.3.3. Configuring SNMP alerts 55
4.3.3.4. Querying SCB status information using agents 56
4.3.4.1. Configuring monitoring 58
4.3.4.3. Preventing disk space fill up 59
4.3.5.1. Creating configuration and data backups 62
4.3.5.2. Archiving the collected data 66
4.3.5.3. Encrypting configuration backups with GPG 69
4.3.5.4.1. Configuring Rsync over SSH 70
4.3.5.4.2. Configuring SMB 72
4.3.5.4.3. Configuring NFS 73
4.4.1. Managing SCB users locally 75
4.4.2. Setting password policies for local users 76
4.4.3. Managing local usergroups 78
4.4.4. Managing SCB users from an LDAP database 79
4.4.5. Authenticating users to a RADIUS server 82
4.4.6.1. Modifying group privileges 85
4.4.6.2. Creating new usergroups for the SCB web interface 86
4.5.1.1. Disabling controlled traffic 92
4.5.1.2. Disabling controlled traffic permanently 93
4.5.2.3. Configuring redundant Heartbeat interfaces 98
4.5.2.5. Configuring next-hop router monitoring 101
4.5.3.1.1. Updating SCB and managing the firmware 104
4.5.3.1.2. Upgrading both the core and the boot firmware of a high availability system 105
4.5.3.1.3. Reverting to an older firmware version 106
4.5.3.2. Updating the SCB license 106
4.5.3.3. Exporting the configuration of SCB 107
4.5.3.4. Importing the configuration of SCB 109
4.5.4.2. Enabling SSH access to the SCB host 111
4.5.4.3. Changing the root password of SCB 112
4.5.5.1. Disabling sealed mode 114
4.5.7.1. Generating certificates for SCB 117

4.5.7.2. Uploading external certificates to SCB 118
5.1.1. Configuring connections 120
5.1.2. Modifying the destination address 123
5.1.3. Modifying the source address 125
5.1.4. Creating and editing channel policies 126
5.1.5. Configuring time policies 129
5.1.6. Creating and editing user lists 130
5.1.7. Authenticating users to an LDAP server 131
5.1.8.1. Encrypting audit trails 135
5.1.8.2.1. Built-in timestamping service 138
5.1.8.2.2. External timestamping service 139
5.1.8.3. Digitally signing audit trails 140
5.1.8.4. Limiting audit trails 141
5.1.9. Verifying certificates with Certificate Authorities 142
5.1.10. Signing certificates on-the-fly 143
5.1.11. Forwarding traffic to an IDS or DLP system 144
5.1.12. Configuring cleanup for the SCB connection database 147
5.2.1. Setting the SSH host keys and certificates of the connection 149
5.2.3.1. Creating a new authentication policy 155
5.2.3.2.1. Local client-side authentication 157
5.2.3.4. How to map keys and certificates 159
5.2.4.1. Automatically adding the host keys and host certificates of a server to SCB 161
5.2.4.2. Manually adding the host key or host certificate of a server 162
5.2.5. Creating and editing protocol-level SSH settings 163
5.3.2. Creating and editing protocol-level RDP settings 166
5.3.3. Joining SCB into a domain 168
5.3.4. Using SSL-encrypted RDP connections 169
5.3.5. Verifying the certificate of the RDP server in encrypted connections 170
5.4.1. Creating and editing protocol-level Telnet settings 172
5.5.1. Creating and editing protocol-level VNC settings 173
6.1.1. Displaying information about connections 176
6.4.2. Creating predefined filters 185
6.5. Configuring custom reports 185
6.7. Configuring full-text indexing of audit trails 189
7.1. Installing the Audit Player application 191
7.2.1. Downloading audit trails from SCB 194
7.2.2. Replaying a session with the Audit Player 195
7.2.3. Replaying SCP and SFTP sessions 197
7.3.3.1. Importing certificates with MMC 202
7.3.3.2. Converting certificates using OpenSSL 203
7.3.3.3. Converting certificates using Firefox 204
7.3.5. Adding a new font to the OCR database 204
7.4.1.1. AP logging in user mode 205
7.4.1.2. AP logging as an indexer service 206
8.1. Configuring usermapping policies 208
8.2.1. Configuring outband gateway authentication 210
8.2.2. Performing outband gateway authentication on SCB 212
8.2.3. Performing inband gateway authentication in SSH connections 213

8.2.4. Performing inband gateway authentication in RDP connections 214
8.3.1. Configuring 4-eyes authorization 215
8.3.2. Performing 4-eyes authorization on SCB 218
9.1.1. Configuring public-key authentication using local keys 220
9.1.2. Configuring public-key authentication using an LDAP server and a fixed key 221
9.1.3. Configuring public-key authentication using an LDAP server and generated keys 222
9.2.1. Organizing connections based on port numbers 223
9.2.2. Organizing connections based on alias IP addresses 223
9.3. Using nontransparent Bastion mode 224
9.4. Restoring SCB configuration and data 226
10.1. SSH usermapping and keymapping in AD with public key 227
11.1. Network troubleshooting 235
11.3. Viewing logs on SCB 236
11.4. Changing log verbosity level of SCB 237
11.5. Collecting logs and system information for error reporting 238
11.6.1. Displaying custom connection statistics 241
11.7.1. Recovering SCB if both nodes broke down 242
11.7.2. Recovering from a split brain situation 243
11.7.3. Replacing a HA node 245
11.8. Restoring configuration from a configuration backup 245
3.1. Installing the SCB hardware 250
3.2. Installing two SCB units in HA mode 251
4.2. Installing SCB under VMware ESXi 253
5.1. Configuring advanced routing on Linux 255

Preface

Welcome to the BalaBit Shell Control Box 3 LTS Administrator Guide!

This document describes how to configure and manage the BalaBit Shell Control Box (SCB). Background information for the technology and concepts used by the product is also discussed.

1. Summary of contents

Chapter 1, Introduction (p. 1) describes the main functionality and purpose of the BalaBit Shell Control Box.

Chapter 2, The concepts of SCB (p. 4) discusses the technical concepts and philosophies behind SCB.

Chapter 3, The Welcome Wizard and the first login (p. 23) describes what to do after assembling SCB — it is a step-by-step guide for the initial configuration.

Chapter 4, Configuring and managing SCB (p. 41) provides detailed description on configuring and managing SCB as a host.

Chapter 5, Configuring connections (p. 120) discusses how to configure and audit SSH, RDP, and Telnet connections.

Chapter 7, Viewing session information and replaying audit trails (p. 191) describes how to search and display audit trails, generate reports, and replay the audited sessions.

Chapter 8, Advanced authentication and authorization techniques (p. 208) describes how to configure gateway authentication and 4-eyes authorization for the connections.

Chapter 9, Best practices and configuration examples (p. 220) gives step-by-step procedures to configure special aspects of SCB.

Chapter 10, SCB scenarios (p. 227) discusses common scenarios for SCB.

Chapter 11, Troubleshooting SCB (p. 235) describes troubleshooting and maintenance procedures of BalaBit Shell Control Box (SCB).

Appendix 1, About the Secure Shell protocol in a nutshell (p. 247) is a brief introduction into the Secure Shell protocol.

Appendix 2, Package contents inventory (p. 249) lists the contents of the package you receive with the BalaBit Shell Control Box.

Appendix 3, BalaBit Shell Control Box Hardware Installation Guide (p. 250) describes how to set up the BalaBit Shell Control Box (SCB) hardware.

Appendix 4, BalaBit Shell Control Box VMware Installation Guide (p. 253) describes how to install BalaBit Shell Control Box (SCB) as a virtual appliance.

Appendix 6, BalaBit Shell Control Box End User License Agreement (p. 257) includes the text of the End User License Agreement applicable to SCB products.

Appendix 7, Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd) License (p. 263) includes the text of the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd) License applicable to The BalaBit Shell Control Box 3 LTS Administrator Guide.

The *Glossary (p. 268)* provides definitions of important terms used in this guide.

The *Index (p. 272)* provides cross-references to important terms used in this guide.

2. Target audience and prerequisites

This guide is intended for auditors, consultants, and security experts responsible for securing, auditing, and monitoring server administration processes, especially remote server management. It is also useful for IT decision makers looking for a tool to improve the security and auditability of their servers, or to facilitate compliance to the Sarbanes-Oxley (SOX), the Health Insurance Portability and Accountability Act (HIPAA), Basel II, or the Payment Card Industry (PCI) standard.

The following skills and knowledge are necessary for a successful SCB administrator:

- At least basic system administration knowledge.
- An understanding of networks, TCP/IP protocols, and general network terminology.
- Working knowledge of the UNIX or Linux operating system is not mandatory but highly useful.
- In-depth knowledge of various server applications is required for forensics situations.

3. Products covered in this guide

This guide describes the use of the BalaBit Shell Control Box version 3 LTS.

4. Typographical conventions

Before you start using this guide, it is important to understand the terms and typographical conventions used in the documentation. For more information on specialized terms and abbreviations used in the documentation, see the Glossary at the end of this document.

The following kinds of text formatting and icons identify special information in the document.

Tip
Tips provide best practices and recommendations.

Note
Notes provide additional information on a topic, and emphasize important facts and considerations.

Warning
Warnings mark situations where loss of data or misconfiguration of the device is possible if the instructions are not obeyed.

`Command`	Commands you have to execute.
Emphasis	Reference items, additional readings.
`/path/to/file`	File names.
`Parameters`	Parameter and attribute names.
Label	GUI output messages or dialog labels.
Menu	A submenu or menu item in the menu bar.
Button	Buttons in dialog windows.

5. Contact and support information

This product is developed and maintained by BalaBit IT Security Ltd. We are located in Budapest, Hungary. Our address is:

BalaBit IT Security Ltd.
2 Alíz Street
H-1117 Budapest, Hungary
Tel: +36 1 398-6700
Fax: +36 1 208-0875
E-mail: <info@balabit.com>
Web: *http://www.balabit.com/*

5.1. Sales contact

You can directly contact us with sales related topics at the e-mail address <sales@balabit.com>, or *leave us your contact information and we call you back*.

5.2. Support contact

To access the BalaBit Online Support System (BOSS), sign up for an account at *the MyBalaBit page* and *request access to the BalaBit Online Support System (BOSS)*. Online support is available 24 hours a day.

BOSS is available only for registered users with a valid support package.

Support e-mail address: <support@balabit.com>.

Support hotline: +36 1 398 6700 (available from 9 AM to 5 PM CET on weekdays)

5.3. Training

BalaBit IT Security Ltd. holds courses on using its products for new and experienced users. For dates, details, and application forms, visit the *http://www.balabit.com/support/trainings/* webpage.

6. About this document

This guide is a work-in-progress document with new versions appearing periodically.

The latest version of this document can be downloaded from the BalaBit website *here*.

6.1. Feedback

Any feedback is greatly appreciated, especially on what else this document should cover. General comments, errors found in the text, and any suggestions about how to improve the documentation is welcome at <documentation@balabit.com>.

Chapter 1. Introduction

This chapter introduces the BalaBit Shell Control Box (SCB) in a non-technical manner, discussing how and why is it useful, and what additional security it offers to an existing IT infrastructure.

1.1. What SCB is

SCB is a device that controls, monitors, and audits remote administrative access to servers. It is a tool to oversee server administrators and server administration processes by controlling the encrypted connections used in server administration. It is an external, fully transparent device, completely independent from the clients and the servers. The server- and client applications do not have to be modified in order to use SCB — it integrates smoothly into the existing infrastructure.

SCB logs all administrative traffic (including configuration changes, executed commands, and so on) into audit trails. All data is stored in encrypted, timestamped and signed files, preventing any modification or manipulation. In case of any problems (server misconfiguration, database manipulation, unexpected shutdown) the circumstances of the event are readily available in the audit trails, thus the cause of the incident can be easily identified. The recorded audit trails can be displayed like a movie – recreating all actions of the administrator. All audit trails are indexed on a separate indexing-server, enabling fast forwarding during replay, searching for events (for example mouse clicks, pressing the Enter key) and texts seen by the administrator. Reports and automatic searches can be configured as well. To protect the sensitive information included in the communication, the two directions of the traffic (client-server and server-client) can be separated and encrypted with different keys, thus sensitive information like passwords are displayed only when necessary.

SCB can also remove the encryption from the traffic and forward the unencrypted traffic to an Intrusion Detection System (IDS), making it possible to analyze the contents of the encrypted traffic. That way traffic that was so far unaccessible for IDS analyzes can be inspected real-time. Other protocols tunneled in SSH can be inspected as well. Similarly, the list of files transferred and accessed in the encrypted protocols can be sent to a Data Leakage Prevention (DLP) system.

SCB has full control over the SSH, RDP, Telnet, TN3270, and VNC connections, giving a framework (with solid boundaries) for the work of the administrators. The most notable features of SCB are the following:

- Disable unwanted channels and features (for example TCP port forwarding, file transfer, VPN, and so on)
- Enforce the use of the selected authentication methods (password, publickey, and so on)
- Require outband authentication on the SCB gateway
- Enforce 4-eyes authorization with real-time monitoring and auditing capabilities
- Audit the selected channels into encrypted and timestamped , and digitally signed audit trails
- Retrieve group memberships of the user from an LDAP database
- Verify the hostkeys and host certificates of the accessed servers

SCB is configured and managed from any modern web browser that supports HTTPS connections, JavaScript, and cookies.

1.2. What SCB is not

SCB is not a firewall. Although it uses advanced firewall technologies, it is an access controlling and auditing device focusing on server administration processes. Actually, it is a device that controls, monitors and audits remote administrative access to servers.

SCB monitors only the passing traffic of administrators accessing the servers remotely. Consequently, it cannot protect the server from local access, nor can it detect such events. If someone has access to a protected server from a local console, then anything he does is beyond the capabilities of SCB.

SCB can be used to control administrative access to the servers. In case of large server farms, it provides a simple way to change or restrict access policies, for example, to disable password-based authentication in SSH, control RDP channels, or to deny the account of an administrator, without having to modify the configuration of each server one-by-one. However, SCB does not and should not be used to replace the proper configuration of the servers, as perfunctory server configuration inevitably introduces security risk beyond the scope of SCB.

1.3. Why is SCB needed?

Server administration must be audited in order to record all important events about a server. However, — for security reasons — servers are almost exclusively administered using encrypted protocols, making system administration difficult to monitor and audit. To achieve reliable auditing, the data collection has to be transparent and independent from the client and the server. Otherwise, a skilled administrator (or attacker) could manipulate the logs to mask the traces of his actions or other events. SCB solves exactly these problems by transparently monitoring the encrypted channels used in administration and introducing a separate auditor layer to oversee system administrators.

The RDP (including VMware View) and VNC-auditing capabilities of SCB is beneficial to record and archive the actions performed on thin-client applications, and helpdesks.

The Telnet-auditing capability of SCB is useful to record and archive the administration of networking devices.

1.4. Who uses SCB?

SCB is useful for everyone who has a server and has to control and audit the activities of the administrators. In particular, SCB is invaluable for:

- *Policy compliance*: Certain regulations — such as the Sarbanes-Oxley Act (SOX) or Basel II — require the financial director of an organization to certify that all financial data they provide to the authorities is accurate and has not been modified. Other industries have similar regulations (like the Health Insurance Portability and Accountability Act (HIPAA), or the Payment Card Industry (PCI)) about protecting personal or credit card information. Such data is usually stored in a database on a central server, and is accessible only via dedicated applications, such as the accounting software. These applications always create the logs and reports necessary for policy compliance. However, these applications are aware only of legitimate accesses to the database. The server storing the database has to be accessible also by server administrators for maintenance reasons. Having superuser privileges on the server, these administrators have the possibility to directly access and manipulate the database, and also to erase the traces of such actions from the server logs. However, SCB can audit the actions of the administrators, complementing the logs and reports of other applications.
- *Organizations having outsourced IT*: Many organizations hire external companies to configure, maintain, and oversee their servers and IT services. This essentially means that the organization is willing to trust the

administrators of this external company with all their data (for example private and business e-mails, customer information, and so on), or even with business-critical services like the operation of their online shop. Obviously, in such situations it is reassuring to have an independent device that can reliably log all administrative activities. SCB does exactly this — it provides detailed information about any problems with the server, making it easy to find those responsible. Using the 4-eyes authorization method, SCB provides real-time control over the remote server access and the administrative actions.

- *Organizations offering remote management*: Organizations on the other end of the outsourcing line — like server- and webhosting companies — can equally benefit from SCB. It gives them the possibility to oversee and audit the administrators, and is also a great tool to evaluate their effectiveness. The recorded audit trails can also be used as evidence to settle any issues about the remotely administered servers. SCB also improves the control over Service Level Agreements (SLA), as the fulfillment of the services can be verified using the recorded audit trails and access reports.
- *Organizations using thin-client infrastructures*: SCB can audit the channels used in popular thin-client solutions, providing an application-independent way to record and monitor the activities of every client.
- *Real-time file transfer and file access monitoring*: SCB can send the list of accessed and transferred files to an external Data Leakage Prevention (DLP) system that can recognize, track and alert on the access (data at rest) and transfer (data in motion) of sensitive data. That way the DLP policy of the organization can be extended to the – so far uncontrolled – encrypted protocols like SSH and SFTP.
- *Organizations in need to control SSH*: Many organizations have to permit outgoing SSH connections, but do not wish to do so without control, as virtually any other protocol can be tunneled into SSH. SCB can control what type of traffic is permitted in an SSH connection, and can separately enable the different traffic types like terminal connections, SFTP file transfers, port- and X11 forwarding, or agent-forwarding.
- *Organizations using jump hosts*: Many organizations use jump hosts to access remote servers or services. SCB can be used to authenticate and audit every access to the jump hosts. Since SCB can supports strong authentication methods (for example, X.509 certification based authentication) and authentication to user directories (for example Microsoft Active Directory and other LDAP databases), it can greatly simplify the key and password management of the hosts. This is especially useful if an organization has to access very many remote hosts, or has lots of jump hosts.

1.5. Public references for BalaBit Shell Control Box:

The following is a list of public references — companies who use SCB in their production environment:

- *Dubai Islamic Bank PJS*
- *FIDUCIA IT AG*
- *Hungarian Post Co. Ltd.*
- Raiffeisen Süd Tirol (*http://www.raiffeisen.it/*)
- *Svenska Handelsbanken AB*
- *Wilo AG*

Chapter 2. The concepts of SCB

This chapter discusses the technical concepts of SCB.

2.1. The philosophy of SCB

SCB is a device that examines Secure Shell (SSH, including forwarded X11 traffic), Secure Copy (SCP), SSH File Transfer Protocol (SFTP), Remote Desktop (RDP), Telnet, VMware View, and VNC connections, ignoring and simply forwarding all other type of traffic. SCB uses man-in-the-middle techniques to decrypt and terminate the inspected connections. It separates the connections into two parts (client — SCB, SCB — server) and inspects all traffic, so that no data can be directly transferred between the server and the client.

Note
SCB is built on application level technology, especially the SSH, RDP, and Telnet proxies of the Zorp Application Level Gateway. For more information on Zorp and this technology, visit *BalaBit's* website.

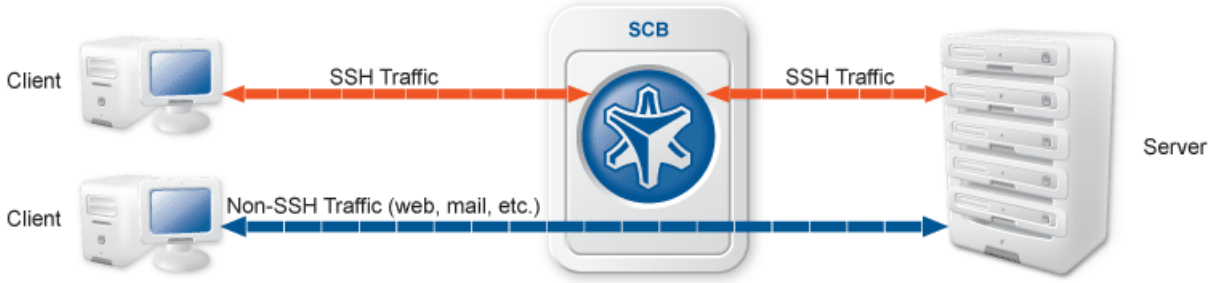


Figure 2.1. Inspecting SSH traffic with SCB

The traffic is inspected on the application level, that is Layer 7 or the application layer of the OSI model. All communication must conform to the standards of the respective protocol, SCB rejects all protocol violations and anomalies to protect the servers from attacks. That way the servers are protected even from unknown attacks exploiting vulnerabilities of the server applications.

SCB has full control over the initial negotiation phase of the connection, when the client and the server decide the parameters of the encryption to be used in the communication. SCB can restrict the use of the various algorithms, forbidding the use of weak ones — an effective shield against downgrade attacks.

Since SCB isolates the client-server connection into two separate connections, the permitted algorithms can be different on the client and the server side.

SCB controls the connections right from the beginning — including user authentication. That way it is easy to mandate strong authentication for protocols where user information is available (for example SSH), because SCB can limit the allowed authentication methods and also the users permitted to access the servers.

SCB uses various policies to restrict who, when, and how can access a connection or a specific channel of the protocol. These policies (based on username, authentication method used, and so on) can be applied to connections between particular clients and servers, or also to specific channels of a connection (for example only to terminal-sessions in SSH, or desktop-sharing in RDP).

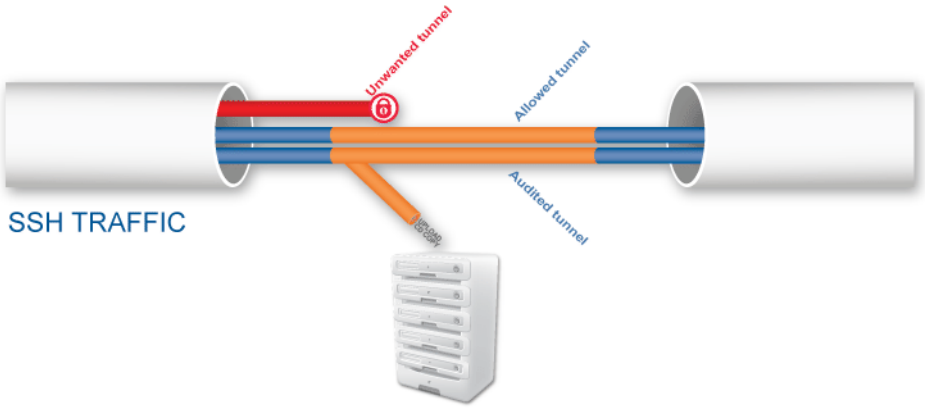


Figure 2.2. Controlling protocol channels

SCB is configured by an administrator or auditor using a web browser.

2.2. Supported protocols and client applications

SCB supports the following protocols and clients. As a general rule, client applications not specifically tested but conforming to the relevant protocol standards should work with SCB.

- *Secure Shell Protocol*: SCB supports only the SSHv2 protocol (RFCs 4251-54). The older and insecure v1 version is not supported. Supported client and server applications include: OpenSSH, Tectia®, and PuTTY.

- *Remote Desktop Protocol*: RDP versions 4-7 are supported. Supported client and server applications: the built-in applications of the Windows XP, Windows Server 2003, Windows Vista, Windows 2008 Server, and Windows 7 platforms.

- *Telnet*: Telnet traffic must conform RFC 854, and to various extensions described in RFCs 856-861, 652-658, 698, 726-27, 732-736, 749, 779, 885, 927, 933, 1041, 1043, 1053, 1073, 1079, 1091, 1096-97, 1184, 1372, 1408, 1572, 2066, 2217, 2840, 2941, and 2946.

  TN3270: Telnet 3270 and 3270E terminal protocols.

- *Virtual Network Computing*: VNC versions 3.3-3.8 are supported. Supported client and server applications: RealVNC, UltraVNC, TightVNC, KVM, Vino.

- *VMware View*: VMware View Clients using the Remote Desktop (RDP) display protocol to access remote servers are supported. For details, see *Section 5.6, VMware View connections (p. 173)*.

2.3. Modes of operation

SCB has three modes of operation: *Bridge*, *Router*, and *Bastion*. Which one is used depends on the layout of the network.

Tip
It is recommended to design the network topology so that only management and server administration traffic passes SCB. This ensures that the services and applications running on the servers are accessible even in case SCB breaks down, so SCB cannot become a single point of failure.

SCB supports also a *nontransparent* mode that works with any of the above three operation modes, but is most often used together with Bastion mode.

2.3.1. Bridge mode

In Bridge mode, SCB acts as a network switch, and connects the network segment of the administrators to the segment of the protected servers at the data link layer (Layer 2 in the OSI model).

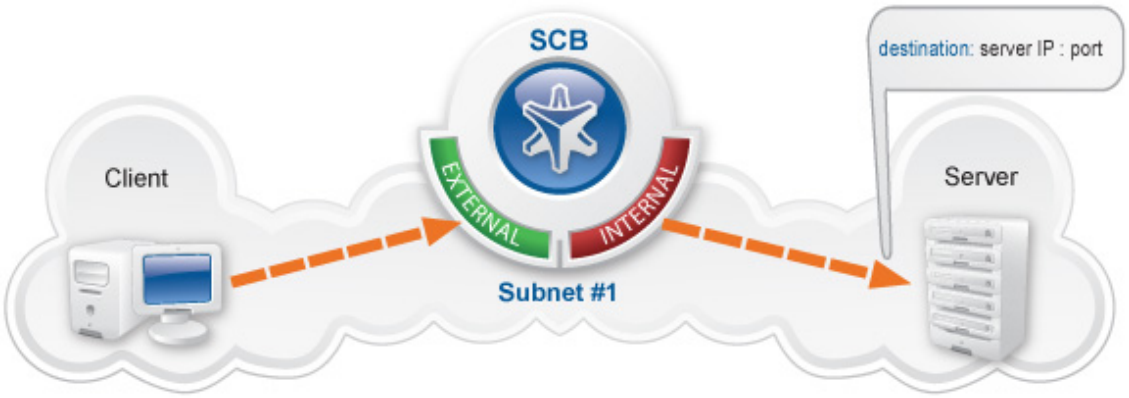


Figure 2.3. SCB in Bridge mode

All traffic passing SCB originates from the MAC address of SCB's bridge interface. SCB replaces all MAC addresses in the ARP replies with its own MAC address. That way all traffic must pass SCB.

Note
In Bridge mode, SCB forwards DHCP traffic between the external and internal interfaces. However, other multicast and broadcast traffic is blocked.

Use Layer 3 devices (routers or firewalls) on both sides of SCB.

Using Bridge mode is recommended in the following network topologies:

Figure 2.4. Recommended topologies for Bridge mode

or

Figure 2.5. Recommended topologies for Bridge mode

Bridge mode in SCB uses a routing table. Use the routing table if more than one subnets are connected to SCB on the opposite side of the default gateway. For details, see *Procedure 4.3.1.2, Routing management traffic to the management interface (p. 49)*.

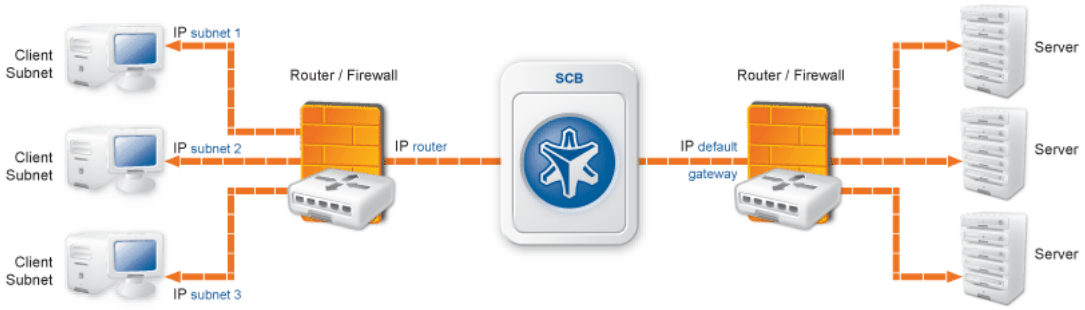


Figure 2.6. When routing is needed in Bridge mode

Warning
Use dedicated switches on both sides of SCB when using bridge mode and high-availablity together. Do not connect other devices to these switches.

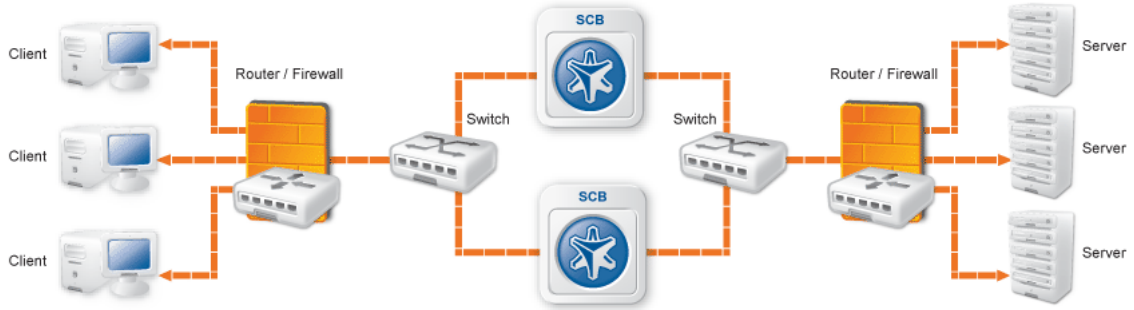


Figure 2.7. Using Bridge mode with high availability

2.3.2. Router mode

In Router mode, SCB acts as a transparent router connecting the network segment of the administrators to the segment of the protected servers at the network layer (Layer 3 in the OSI model). All connections must pass SCB to reach the servers — SCB is a proxy gateway, completely separating the protected servers from the rest of the network. Controlled connections and traffic are inspected on the application level, while other types of connections are simply forwarded on the packet level.

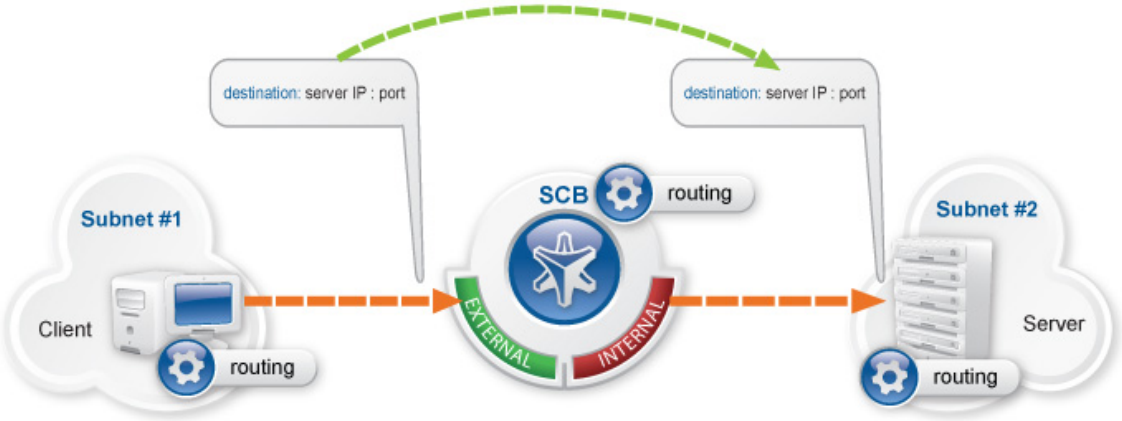


Figure 2.8. SCB in Router mode

Warning
Routing mode does not support multicast traffic.

Tip
In case you use SCB in Router mode, but would like both the client-side and server-side traffic to go through the external interface, set a nonexistent IP address to the internal interface (that is, an address that does not exist on the network). In this case SCB will route the traffic back to the default gateway through the external interface.

2.3.3. Single-interface router mode

Single-interface router mode is similar to *Router mode*, but both the client-side and server-side traffic uses the same interface. An external device — typically a firewall or a router — is required that actively redirects the audited traffic to SCB. To accomplish this, the external device must support advanced routing (also called policy-based routing). For details on configuring an external Linux-based router, see *Procedure 5.1, Configuring advanced routing on Linux (p. 255)*.

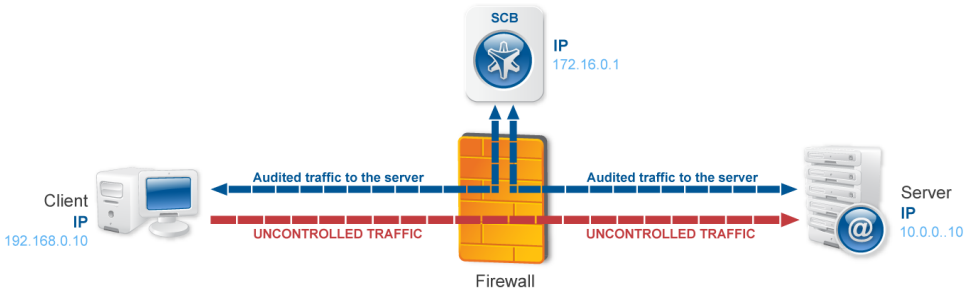


Figure 2.9. SCB in single-interface router mode

Advantages

The advantages of using the single-interface router mode are:

- Totally transparent for the clients, no need to modify their configuration
- The network topology is not changed
- Only the audited traffic is routed to SCB, production traffic is not

Disadvantages

The disadvantages of using the single-interface router mode are:

- If configured to do so, SCB performs IP spoofing. This might be unacceptable in certain environments
- Traffic must be actively routed to SCB using an external device, consequently a network administrator can disable SCB by changing routing rules
- When adding a new port or subnet to the list of audited connections, the configuration of the external device must be modified as well.

2.3.4. Bastion mode

In this mode SCB acts as a bastion host — administrators can address only SCB, the administered servers cannot be targeted directly. The firewall of the network has to be configured to ensure that only connections originating from SCB can access the servers. SCB determines which server to connect based on the parameters of the incoming connection (the IP address of the administrator and the target IP and port).

Bastion mode inherently ensures that only the controlled (management and server administration) traffic reaches SCB. This ensures that the services and applications running on the servers are accessible even in case SCB breaks down, so SCB cannot become a single point of failure. The firewall or router before the servers must be configured to direct the controlled SSH, RDP, Telnet, and VNC traffic to SCB. This can be accomplished on the IP level.

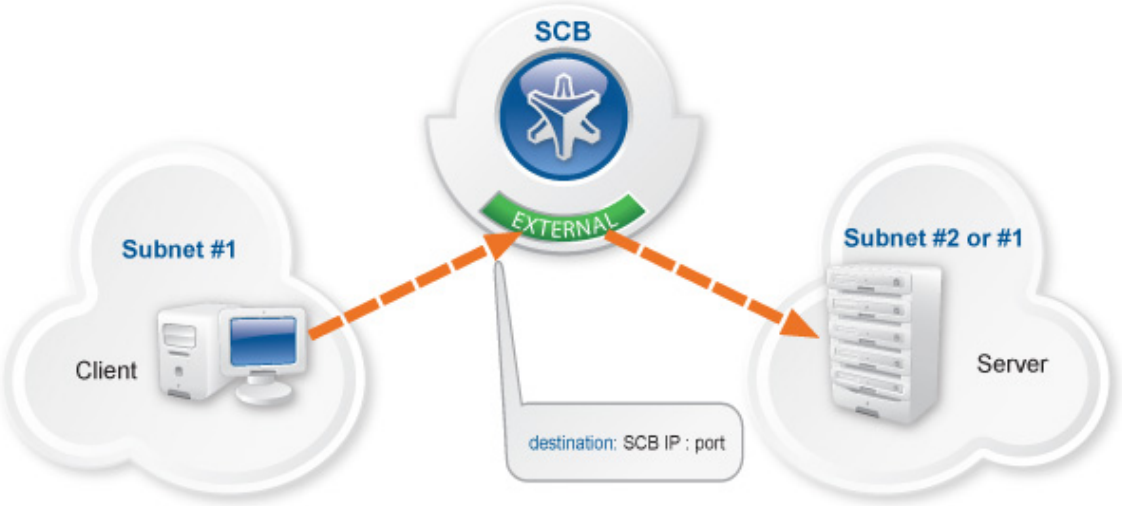


Figure 2.10. SCB in Bastion mode

Tip
Bastion mode is useful if the general (not inspected) traffic is very high and could not be forwarded by SCB.

Bastion mode is often used together with the Nontransparent mode (for details, see *Section 2.3.5, Nontransparent mode (p. 11)*).

2.3.5. Nontransparent mode

Nontransparent mode allows you to create a single connection policy and allow users to access any server by including the name of the target server in their username (for example `ssh username@targetserver:port@scb_address`). SCB can extract the address from the username and direct the connection to the target server.

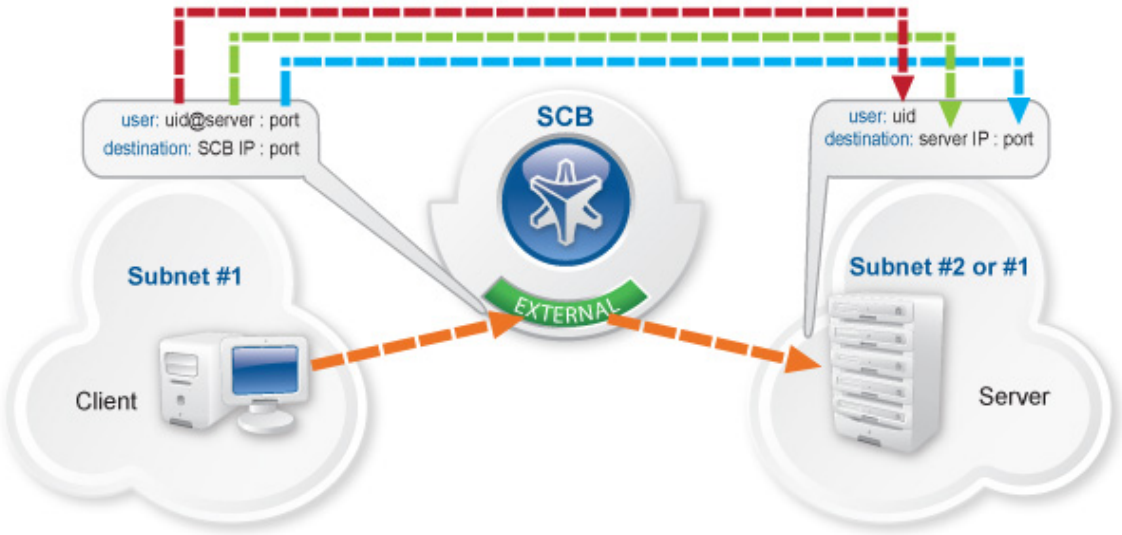


Figure 2.11. SCB in Nontransparent mode

Since some client applications do not permit the `@` and `:` characters in the username, therefore alternative characters can be used as well:

- To separate the username and the target server, use the `@` or `%` characters, for example: `username%targetserver@scb_address`
- To separate the target server and the port number, use the `:`, `+`, or `/` characters, for example: `username%targetserver+port@scb_address`

Windows RDP clients often send only the first 9 characters of the username to the server, making nontransparent operation difficult. When using the RDP4 or RDP5 protocols, it is not necessary to include the username when starting an RDP connection (for example use only `@server`); the user can type the username later in the graphical login screen. However, the username must be specified for RDP6 connections.

2.4. Connecting to a server through SCB

When a client initiates a connection to a server, SCB performs one of the procedures detailed below, depending on the protocol used in the connection.

- For SSH connections, see *Procedure 2.4.1, Connecting to a server through SCB using SSH (p. 12)*.
- For RDP and other connections, see *Procedure 2.4.2, Connecting to a server through SCB using RDP (p. 227)*.

2.4.1. Procedure – Connecting to a server through SCB using SSH

Steps:

Step 1. The client tries to connect the server. SCB receives the connection request and establishes the TCP connection with the client.

Step 2. SCB examines the connection request: it checks the IP address of the client and the IP address and port number of the intended destination server. If these parameters of the request match with a connection policy configured on SCB, SCB inspects the connection in detail. Other connections are ignored by SCB; they are simply forwarded on the packet level.

The selected connection policy determines all settings and parameters of the connection.

Note
SCB compares the connection policies to the parameters of the connection request one-by-one, starting with the first policy in the policy list. The first connection policy completely matching the connection request is applied to the connection. For details, see *Procedure 5.1.1, Configuring connections (p. 120)*.

Step 3. Optionally, before establishing the server-side connection, SCB can evaluate the connection and channel policies to determine if the connection might be permitted at all, for example it is not denied by a Time Policy. For details, see *Procedure 5.2.5, Creating and editing protocol-level SSH settings (p. 163)*.

Step 4. SCB selects the destination server based on the **Target** parameter of the connection policy. Network address translation of the target address can be performed at this step. For details, see *Procedure 5.1.2, Modifying the destination address (p. 123)*.

Step 5. SCB selects the source address used in the server-side connection based on the **SNAT** parameter of the connection policy. For details, see *Procedure 5.1.3, Modifying the source address (p. 125)*.

Step 6. SCB establishes the TCP connection to the server.

Step 7. SCB negotiates the protocol parameters of the connection (for example SSH encryption parameters) according to the **Protocol settings** of the connection policy. The SSH handshake is performed simultaneously on the server- and the client-side.

Step 8. SCB displays an SSH hostkey to the client. This hostkey is either generated on SCB, or it is the hostkey of the server (if it is available on SCB). The connection policy determines the hostkey shown to the client.

Warning
If the SSH Settings of the connection enable only RSA keys in the connection, the RSA key shown to the client must be set in the connection policy. Similarly, if only DSA keys are permitted, the DSA key must be set.

Step 9. SCB verifies the hostkey of the server. If the server has not been contacted before, SCB can accept and store the hostkey of the server. Alternatively, the hostkey of the server can be manually uploaded to SCB. For details, see *Section 5.2.4, Server host keys and certificates (p. 161)*.

Step 10. The client authenticates itself using an authentication method permitted by the **Authentication Policy** set in SCB. Different connections can use different authentication policies, thus allow different authentication methods. The Authentication policy also restricts which users can connect the server if public-key authentication is used.

Step 11. If the authentication is successful, SCB checks whether the particular user is allowed to access the connection by consulting the **User list** assigned to the connection policy in the channel policy. If needed, SCB connects the LDAP servers set in the **LDAP Servers** policy to resolve the group memberships of the user.

Warning
User Lists are white- or blacklists of usernames that determine who can access the server remotely. However, this cannot prevent a user from accessing the server from a local terminal.

Step 12. SCB performs the authentication on the server, using the data received from the client in Step 8.

Step 13. Both the server and the client side connections have been established. From this step, the client can try to open any type and any number of channels in the connection.

Step 14. SCB consults the **Channel Policy** assigned to the connection for each channel opening request to see if the channel type (for example TCP forward or terminal session in SSH) requested by the client is permitted in the connection. Channel policies may impose restrictions on the availability of certain channels.

Step 15. SCB consults the **Time Policy** assigned to the channel policy. Channels may be opened only within the allowed period.

Tip
Time policies are a good way to ensure that the server can be accessed only within the specified timeframe.

Step 16. If gateway authentication is set for the connection policy, the SSH session of the client is paused until the client successfully completes the gateway authentication on SCB. The server-side connection is established only after the gateway authentication is completed. For details, see *Procedure 2.7, The gateway authentication process (p. 16)*.

Step 17. If the Authentication Policy of the SSH connection policy requires separate authentication on the client and the server side, the client performs the client-side authentication.

Step 18. If the Authentication Policy of the SSH connection policy requires separate authentication on the client and the server side, the client performs the server-side authentication.

Step 19. If 4-eyes authorization is set in the Channel Policy, the SSH session of the client is paused until the authorizer permits the client to connect to the server. For details, see *Procedure 2.8, 4-eyes authorization (p. 17)*.

Step 20. The client starts to work on the server. SCB records the entire communication into digitally encrypted audit trails if auditing is enabled in the Channel Policy. For details, see *Procedure 5.1.4, Creating and editing channel policies (p. 126)*.

2.4.2. Procedure – Connecting to a server through SCB using RDP

Steps:

Step 1. The client tries to connect the server. SCB receives the connection request and establishes the TCP connection with the client.

Step 2. SCB examines the connection request: it checks the IP address of the client and the IP address and port number of the intended destination server. If these parameters of the request match with a connection policy configured on SCB, SCB inspects the connection in detail. Other connections are ignored by SCB; they are simply forwarded on the packet level.
The selected connection policy determines all settings and parameters of the connection.

Note
SCB compares the connection policies to the parameters of the connection request one-by-one, starting with the first policy in the policy list. The first connection policy completely matching the connection request is applied to the connection. For details, see *Procedure 5.1.1, Configuring connections (p. 120)*.

Step 3. SCB selects the destination server based on the **Target** parameter of the connection policy. Network address translation of the target address can be performed at this step. For details, see *Procedure 5.1.2, Modifying the destination address (p. 123)*.

Step 4. SCB selects the source address used in the server-side connection based on the **SNAT** parameter of the connection policy. For details, see *Procedure 5.1.3, Modifying the source address (p. 125)*.

Step 5. SCB establishes the TCP connection to the server.

Step 6. SCB checks the protocol parameters of the connection (for example version of the RDP protocol used) according to the **Protocol settings** of the connection policy. The RDP handshake is performed simultaneously on the server- and the client-side.

Step 7. The server opens a Drawing channel for the user to perform authentication.

Step 8. SCB consults the **Channel Policy** assigned to the connection to check if the Drawing channel is enabled.

Step 9. SCB consults the **Time Policy** assigned to the connection. Connections are accepted only within the allowed period.

Tip
Time policies are a good way to ensure that the configuration of the server is modified only during maintenance hours.

Step 10. If gateway authentication is set for the connection policy, the session of the client is paused until the client successfully completes the gateway authentication on SCB. The server-side connection is established only after the gateway authentication is completed. For details, see *Procedure 2.7, The gateway authentication process (p. 16)*.

Step 11. The client authenticates on the server.

Step 12. If 4-eyes authorization is set in the Channel Policy, the session of the client is paused until the authorizer permits the client to connect to the server. For details, see *Procedure 2.8, 4-eyes authorization (p. 17)*.

Step 13. SCB establishes the server-side connection and authenticates the client on the server. If the authentication fails for any reason, SCB terminates the client-side connection as well. This is required to verify the username of the client when it attempts to access the server again.

Step 14. Both the server and the client side connections have been established. From this step, the server can open any type and any number of channels in the connection. (In RDP, the client requests a channel, and the server opens the channels towards the client.) SCB consults the **Channel Policy** assigned to the connection to check if the particular channel is enabled in the connection.

Step 15. The client starts to work on the server. SCB records the entire communication into digitally encrypted audit trails if auditing is enabled in the Channel Policy. For details, see *Procedure 5.1.4, Creating and editing channel policies (p. 126)*.

2.5. SSH hostkeys

The SSH communication authenticates the remote SSH server using public-key cryptography, either using plain hostkeys, or X.509 certificates. Client authentication can also use public-key cryptography. The identity of the remote server can be verified by inspecting its hostkey or certificate. When trying to connect a server via SCB, the client sees a hostkey (or certificate) shown by SCB. This key is either the hostkey of SCB, or the original hostkey of the server, provided that the private key of the server has been uploaded to SCB. In the latter case the client will not notice any difference and have no knowledge that he is not communicating directly with the server, but with SCB.

2.6. Authenticating clients using public-key authentication in SSH

Public-key authentication requires a private and a public key (or an X.509 certificate) to be available on SCB. First, the public key of the user is needed to verify the user's identity in the client-side SSH connection: the key presented by the client is compared to the one stored on SCB. SCB uses a private key to authenticate itself to the sever in the server-side connection. SCB can use the private key of the user if it is uploaded to SCB. Alternatively, SCB can generate a new keypair, and use its private key for the server-side authentication, or use agent-forwarding, and authenticate the client with its own key.

Warning
If SCB generates the private key for the server-side authentication, then the public part of the keypair must be imported to the server; otherwise the authentication will fail. Alternatively, SCB can upload the public key (or a generated X.509 certificate) into an LDAP database.

2.7. Procedure – The gateway authentication process

Purpose:

When gateway authentication is required for a connection, the user must authenticate on SCB as well. This additional authentication can be performed on the SCB web interface, so it provides a protocol-independent, outband authentication method. That way the connections can be authenticated to the central authentication database (for example LDAP or RADIUS), even if the protocol itself does not support authentication databases. Also, connections using general usernames (for example `root`, `Administrator`, and so on) can be connected to real user accounts.

Note
For SSH connections, gateway authentication can be performed also inband, without having to access the SCB web interface. Inband gateway authentication must be performed when client-side authentication is configured. For details on configuring client-side authentication, see *Section 5.2.3.2, Client-side authentication settings (p. 156).*

Figure 2.12. Gateway authentication

Technically, the process of gateway authentication is the following:

Steps:

Step 1. The user initiates a connection from a client.

Step 2. If gateway authentication is required for the connection, SCB pauses the connection.

Step 3. The user logs in to the SCB web interface, selects the connection from the list of paused connections, and enables it. It is possible to require that the authenticated session and the web session originate from the same client IP address.

Step 4. The user performs the authentication on the server.

Note
Gateway authentication can be used together with other advanced authentication and authorization techniques like 4-eyes authorization, client- and server-side authentication, and so on.

2.8. Procedure – 4-eyes authorization

Purpose:

When 4-eyes authorization is required for a connection, a user (called authorizer) must authorize the connection on SCB as well. This authorization is in addition to any authentication or group membership requirements needed for the user to access the remote server. Any connection can use 4-eyes authorization, so it provides a protocol-independent, outband authorization and monitoring method.

The authorizer has the possibility to terminate the connection any time, and also to monitor real-time the events of the authorized connections: SCB can stream the traffic to the Audit Player application, where the authorizer (or a separate auditor) can watch exactly what the user does on the server, just like watching a movie.

Note
The auditor can only see the events if the required decryption keys are available on the host running the Audit Player application.

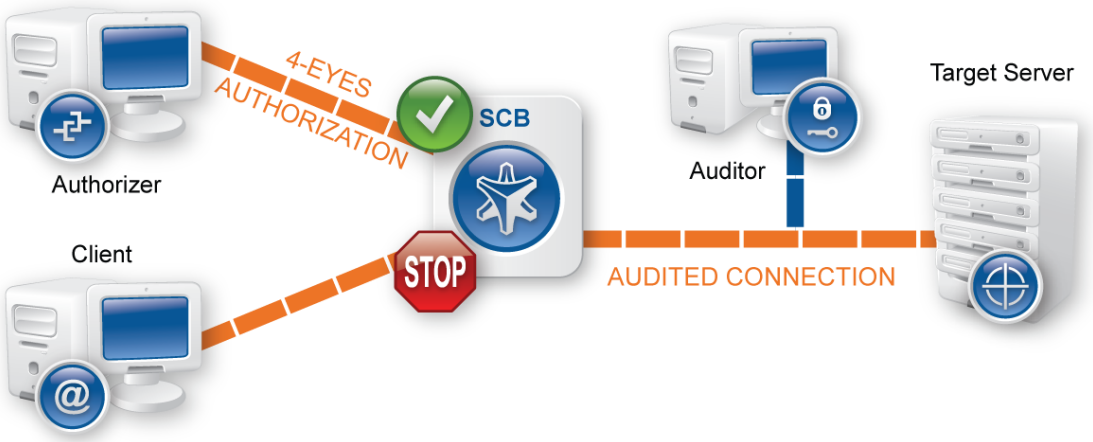


Figure 2.13. 4-eyes authorization

Technically, the process of 4-eyes authorization is the following:

Steps:

Step 1. The user initiates a connection from a client.

Step 2. If 4-eyes authorization is required for the connection, SCB pauses the connection.

Step 3. The authorizer logs in to the SCB web interface, selects the connection from the list of paused connections, and enables it.

Step 4. The user performs the authentication on the server.

Step 5. The auditor (who can be the authorizer, but is it possible to separate the roles) watches the actions of the user real-time.

Note
4-eyes authorization can be used together with other advanced authentication and authorization techniques like gateway authentication , client- and server-side authentication, and so on.

2.9. Network interfaces

The SCB hardware has five network interfaces: the external, the management, the internal, the HA, and the IPMI interface. For details on hardware installation, see *Appendix 3, BalaBit Shell Control Box Hardware Installation Guide (p. 250)*.

The *internal* interface is used exclusively for communication between SCB and the protected servers. The internal interface uses the Ethernet connector labeled as `INT`.

The *external* interface is used for communication between SCB and the clients: the controlled connections of the clients (for example, the SSH connections of the system administrators managing the servers) accessing the protected servers enter via this interface. Also, the initial configuration of SCB is always performed using the external interface (For details on the initial configuration, see *Procedure 3.2, Configuring SCB with the Welcome Wizard (p. 29)*). The external interface is used for management purposes if the management interface is not configured. The external interface uses the Ethernet connector labeled as `EXT`.

The *management* interface is used exclusively for communication between SCB and the auditors or the administrators of SCB. Incoming connections are accepted only to access the SCB web interface, other connections targeting this interface are rejected. The management interface uses the Ethernet connector labeled as `MGMT`.

The routing rules determine which interface is used for transferring remote backups and syslog messages of SCB.

Tip
It is recommended to direct backups, syslog and SNMP messages, and e-mail alerts to the management interface. For details, see *Procedure 4.3.1.2, Routing management traffic to the management interface (p. 49)*.

If the management interface is not configured, the external interface takes the role of the management interface.

The *HA* interface is an interface reserved for communication between the nodes of SCB clusters. The HA interface uses the Ethernet connector labeled as `HA`. For details on high availability, see *Section 2.10, High Availability support in SCB (p. 19)*.

The Intelligent Platform Management Interface (`IPMI`) interface allows system administrators to monitor system health and to manage SCB events remotely. IPMI operates independently of the operating system of SCB.

2.10. High Availability support in SCB

In high availability (HA) mode two SCB units (called master and slave nodes) having identical configuration are operating simultaneously. The master shares all data with the slave node, and if the master node stops functioning, the other one becomes immediately active, so the servers are continuously accessible. The slave node takes over the MAC addresses of the interfaces of the master node.

Warning
Hazard of data loss! If the two nodes cannot communicate, the slave assumes that the master node broke down and becomes active, which can lead to both nodes being active at the same time (a split brain situation). This might result in data loss. For details on how to recover from a split brain situation, see *Procedure 11.7.2, Recovering from a split brain situation (p. 243)*. For details on how to avoid split brain situations, see *Procedure 4.5.2.3, Configuring redundant Heartbeat interfaces (p. 98)*.

The slave node automatically synchronizes its hard disk with the master via the HA network interface. The disks must be synchronized for the HA support to operate correctly.

Warning
When using the management interface and high availability together, do not forget to connect the management interface of both SCB nodes to the network. Otherwise you will not be able to remotely access SCB if a takeover occurs.

2.11. Firmware in SCB

The SCB firmware is separated into two parts: an *external* and an *internal* firmware.

- The *external* firmware (also called boot firmware) boots up SCB, provides the high availability support, and starts the internal firmware. The external firmware changes very rarely.
- The *internal* firmware (also called core firmware) handles everything else: provides the web interface, manages the connections, and so on. The internal firmware is updated regularly as new features are added to SCB.

Both firmwares can be updated from the SCB web interface. For details, see *Section 4.5.3, Upgrading SCB (p. 103)*.

2.11.1. Firmwares and high availability

When powering on the SCB nodes in high availability mode, both nodes boot and start the boot firmware. The boot firmwares then determine which unit is the master: the core firmware is started only on the master node.

Upgrading the SCB firmware via the web interface automatically upgrades the firmware on both nodes.

2.12. Versions and releases of SCB

As of June 2011, the following release policy applies to BalaBit Shell Control Box:

- *Long Term Supported or LTS releases* (for example, SCB 3 LTS) are supported for 3 years after their original publication date and for 1 year after the next LTS release is published (whichever date is later). The second digit of the revisions of such releases is 0 (for example, SCB 3.0.1). Maintenance releases to LTS releases contain only bugfixes and security updates.
- *Feature releases* (for example, SCB 3 F1) are supported for 6 months after their original publication date and for 2 months after succeeding Feature or LTS Release is published (whichever date is later). Feature releases contain enhancements and new features, presumably 1-3 new feature per release. Only the last feature release is supported (for example when a new feature release comes out, the last one becomes unsupported within two months).

For a full description on stable and feature releases, see *Stable and feature releases*.

Warning
Downgrading from a feature release to an earlier (and thus unsupported) feature release, or to the stable release is not supported: this means that once you upgrade a system from a stable release (for example 1.0) to a feature release (for example 1.1), you will have to keep upgrading to the new feature releases until the next stable version release (for example 2.0) is published, or risk using an unsupported product.

2.13. Accessing and configuring SCB

SCB has a web interface and is configured from a browser. The users of SCB can be authenticated using local, LDAP, or RADIUS databases. The privileges of the users are determined by group memberships; this can be managed either locally on SCB, or centrally in an LDAP database. Assigning privileges to groups is based on ACLs. It is also possible to match groups existing in the LDAP database to a set of SCB privileges. The access control in SCB is very detailed, it is possible to define exactly who can access which parts of the interface and of the stored data.

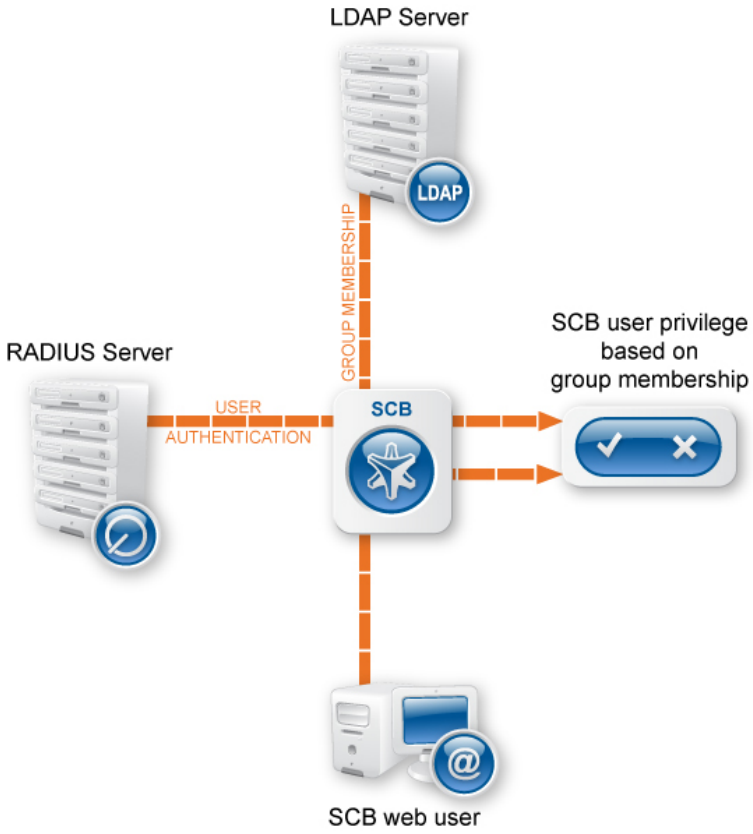


Figure 2.14. Authenticating the users of SCB

2.14. Licenses

SCB's license determines the number of servers (IP addresses) that SCB protects; the license limits the number of IP addresses that SCB can connect on the internal interface.

- *BalaBit Shell Control Box N1000* and *N1000d*: Available with license for 10 IP addresses; can be expanded up to unlimited IP addresses.
- *BalaBit Shell Control Box N10000*: Available with license for 50 IP addresses; can be expanded up to unlimited IP addresses.
- *BalaBit Shell Control Box VA*: Available with license for 5 IP addresses; can be expanded up to unlimited IP addresses.

Contact BalaBit or your local distributor for details. For details on installing a new license, see *Procedure 4.5.3.2, Updating the SCB license (p. 106)*.

Chapter 3. The Welcome Wizard and the first login

This chapter describes the initial steps of configuring SCB. Before completing the steps below, unpack, assemble, and power on the hardware. Connect at least the external network interface to the local network, or directly to the computer from which SCB will be configured.

Note
For details on unpacking and assembling the hardware, see *Appendix 3, BalaBit Shell Control Box Hardware Installation Guide (p. 250)*. For details on how to create a high availability SCB cluster, see *Procedure 3.2, Installing two SCB units in HA mode (p. 251)*.

3.1. The initial connection to SCB

SCB can be connected from a client machine using any modern web browser.

Supported web browsers and operating systems
Supported browsers: Mozilla Firefox 3.0 or newer and Microsoft Internet Explorer 7 and 8. The browser must support HTTPS connections, JavaScript, and cookies. Make sure that both JavaScript and cookies are enabled.

Note
SCB displays a warning message if your browser is not supported or JavaScript is disabled.

Supported operating systems: Microsoft Windows XP, Windows 2003 Server, Windows Vista, Windows 2008 Server, Windows 7, and Linux.

The SCB web interface can be accessed only using SSLv3 or TLSv1 encryption and strong cipher algorithms.

Note that when using Internet Explorer 7 on Windows 2008 to access SCB you must enable active scripting for the Internet Zone, otherwise the SCB web interface will not operate properly. This is not required on other platforms or browser versions. To accomplish this, select **Tools > Internet Options > Security > Internet Zone > Custom level**, and set the **Active scripting** field to `Enabled`.

When using Internet Explorer 8.0 on Windows Server 2008 R2 Enterprise, disable the Content Advisor in the Internet Explorer. To accomplish this, select **Tools > Internet Options > Content > Content Advisor > Disable**.

SCB can be accessed from the local network — it is listening for HTTPS connections on the `192.168.1.1` IP address via its *external* interface (`EXT`, for details on the network interfaces, see *Section 2.9, Network interfaces (p. 18)*). The `192.168.1.0/24` subnet must be accessible from the client. If the client machine is in a different subnet (for example its IP address is `192.168.10.X`), but in the same network segment, the easiest way is to assign an alias IP address to the client machine. Creating an alias IP on the client machine virtually puts both the client and SCB into the same subnet, so that they can communicate. To create an alias IP complete the following steps.

- For details on creating an alias IP on Microsoft Windows, see *Procedure 3.1.1, Creating an alias IP address (Microsoft Windows) (p. 24)*.
- For details on creating an alias IP on Linux, see *Procedure 3.1.2, Creating an alias IP address (Linux) (p. 28)*.

- If configuring an alias interface is not an option for some reason, you can modify the IP address of SCB. For details, see *Procedure 3.1.3, Modifying the IP address of SCB (p. 28)*.

Warning
The Welcome Wizard can be accessed only using the external network interface of SCB, as the management interface is not configured yet.

3.1.1. Procedure – Creating an alias IP address (Microsoft Windows)

Purpose:

This procedure describes how to assign an alias IP address to a network interface on Microsoft Windows platforms.

Steps:

Step 1. Navigate to **Start menu > Settings > Network Connections**.

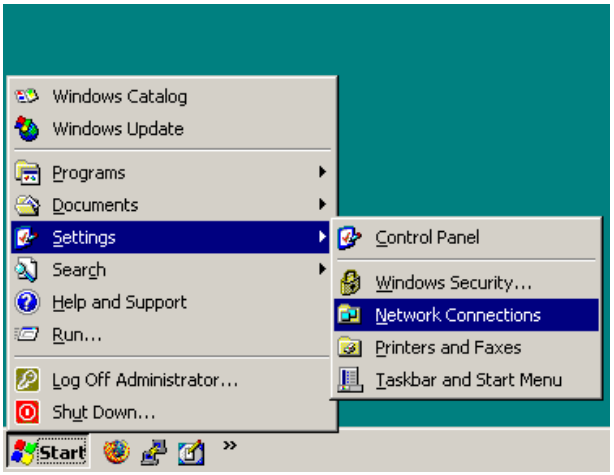


Figure 3.1.

Step 2. Double click on the **Local Area Connection** and then click **Properties**.

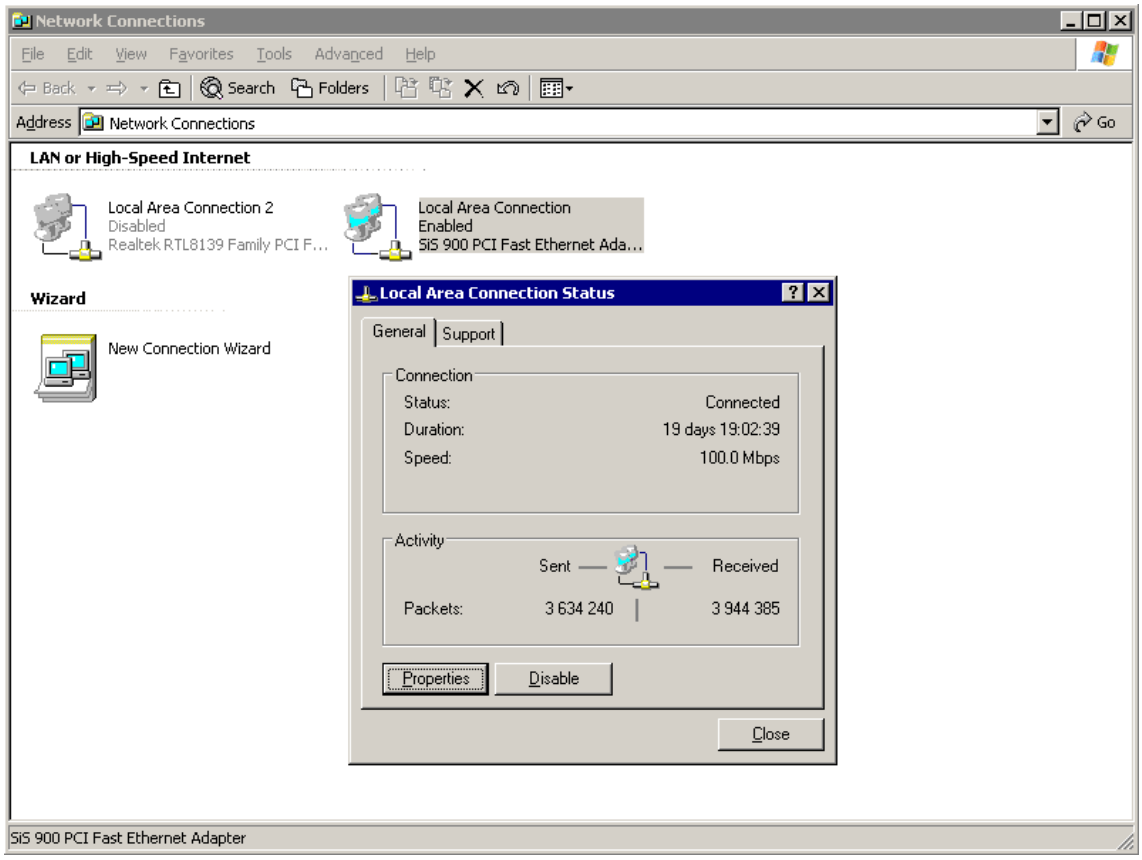


Figure 3.2.

Step 3. Select the **Internet Protocol (TCP/IP)** component in the list and click **Properties**.

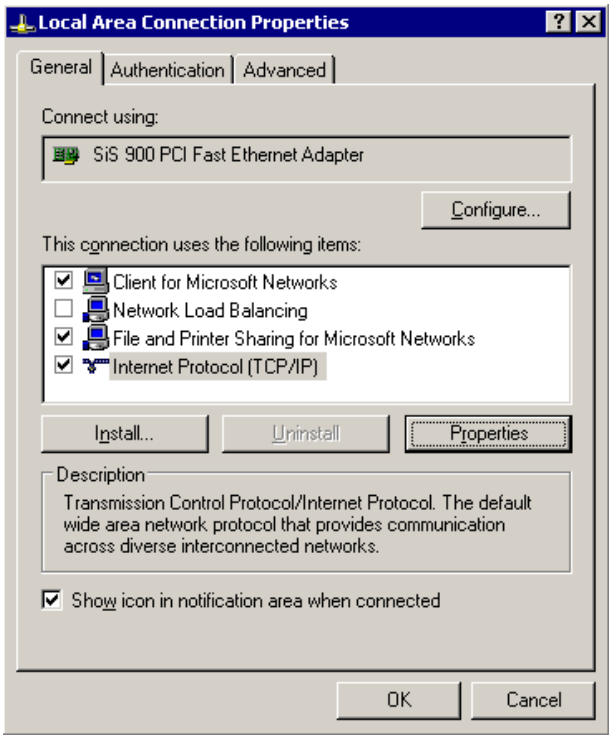


Figure 3.3.

Step 4. To display the Advanced TCP/IP Settings window, click **Advanced**.

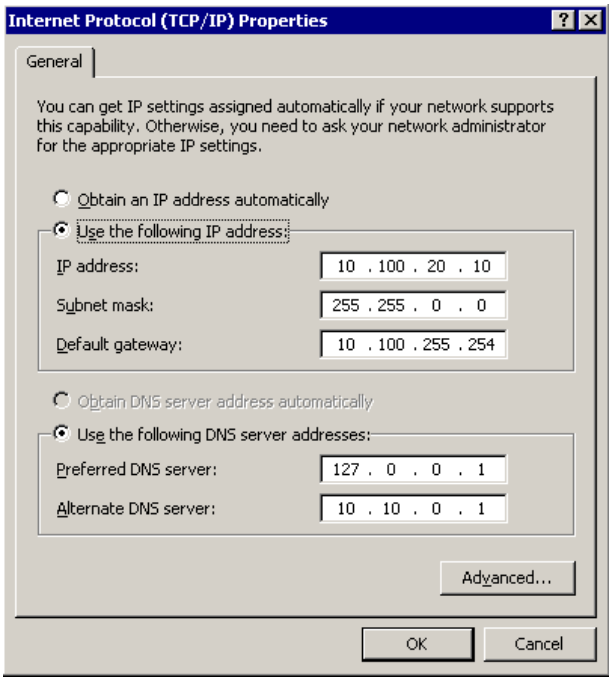

Internet Protocol (TCP/IP) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 10 . 100 . 20 . 10
Subnet mask: 255 . 255 . 0 . 0
Default gateway: 10 . 100 . 255 . 254
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server: 127 . 0 . 0 . 1
Alternate DNS server: 10 . 10 . 0 . 1
Advanced...
OK
Cancel

Figure 3.4.

Step 5. Select the **IP Settings** tab and in the **IP Addresses** section, click **Add**.

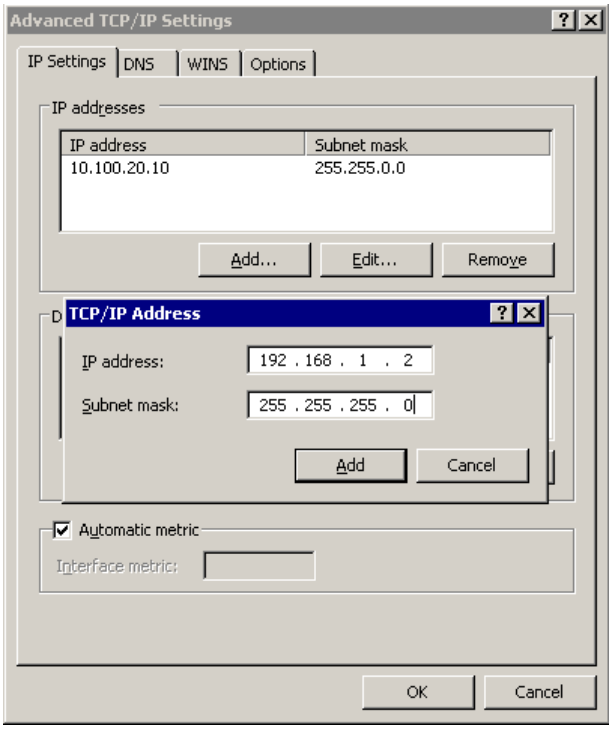


Figure 3.5.

Step 6. Into the **IP Address** field, enter *192.168.1.2*. Into the **Netmask** field, enter *255.255.255.0*.

Warning
If your internal network uses the *192.168.1.0/24* IP range, the *192.168.1.1* and *192.168.1.2* addresses might already be in use. In this case, disconnect SCB from the network, and connect directly a computer to its external interface using a standard cross-link cable.

Step 7. To complete the procedure, click **Add** .

3.1.2. Procedure – Creating an alias IP address (Linux)

Purpose:

This procedure describes how to assign an alias IP address to a network interface on Linux platforms.

Steps:

Step 1. Start a terminal console (for example `gnome-terminal`, `konsole`, `xterm`, and so on).

Step 2. Issue the following command as root:

```
ifconfig <ethX>:0 192.168.1.2
```

where *<ethX>* is the ID of the network interface of the client, usually *eth0* or *eth1*.

Step 3. Issue the `ifconfig` command. The *<ethX>:0* interface appears in the output, having *inet addr:192.168.1.2* .

Step 4. Issue the `ping -c 3 192.168.1.1` command to verify that SCB is accessible. A similar result is displayed:

```
user@computer:~$ ping -c 3 192.168.1.1
                  PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
                  64 bytes from 192.168.1.1: icmp-seq=1 ttl=63 time=0.357 ms

                  64 bytes from 192.168.1.1: icmp-seq=2 ttl=63 time=0.306 ms

                  64 bytes from 192.168.1.1: icmp-seq=3 ttl=63 time=0.314 ms


                  --- 192.168.1.1 ping statistics ---
                  3 packets transmitted, 3 received, 0% packet loss, time
2013ms
                  rtt min/avg/max/mdev = 0.306/0.325/0.357/0.030 ms
```

Open the page *https://192.168.1.1* from your browser and accept the certificate shown. The Welcome Wizard of SCB appears.

3.1.3. Procedure – Modifying the IP address of SCB

Purpose:

To configure SCB to listen for connections on a custom IP address, complete the following steps.

Warning
Use this procedure only before the initial configuration of SCB, that is, before completing the Welcome Wizard. For details on changing the IP address or other network settings of a configured SCB system, see *Section 4.3.1, Network settings (p. 46).*

Steps:

Step 1. Access SCB from the local console.

Step 2. Login using the `root` as username and `default` as password.

Step 3. In the Console Menu, select **Shells > Core shell**.

Step 4. Change the IP address of SCB using the following command. Replace <IP-address> with an IPv4 address suitable for your environment.
```
ifconfig eth0 <IP-address> netmask 255.255.255.0
```

Step 5. Set the default gateway using the following command:
```
route add default gw <IP-address-of-the-default-gateway>
```

Step 6. Type `exit`, then select **Logout** from the Console Menu.

Step 7. Open the page *https://<IP-address-you-set-for-SCB>* from your browser and accept the certificate shown. The Welcome Wizard of SCB appears.

3.2. Procedure – Configuring SCB with the Welcome Wizard

Purpose:

The Welcome Wizard guides you through the basic configuration steps of SCB. All parameters can be modified before the last step by using the **Back** button of the wizard, or later via the web interface of SCB.

Steps:

Step 1. Open the page *https://192.168.1.1* in your browser and accept the displayed certificate. The Welcome Wizard of SCB appears.

Step 2. When configuring SCB for the first time, click **Next**.

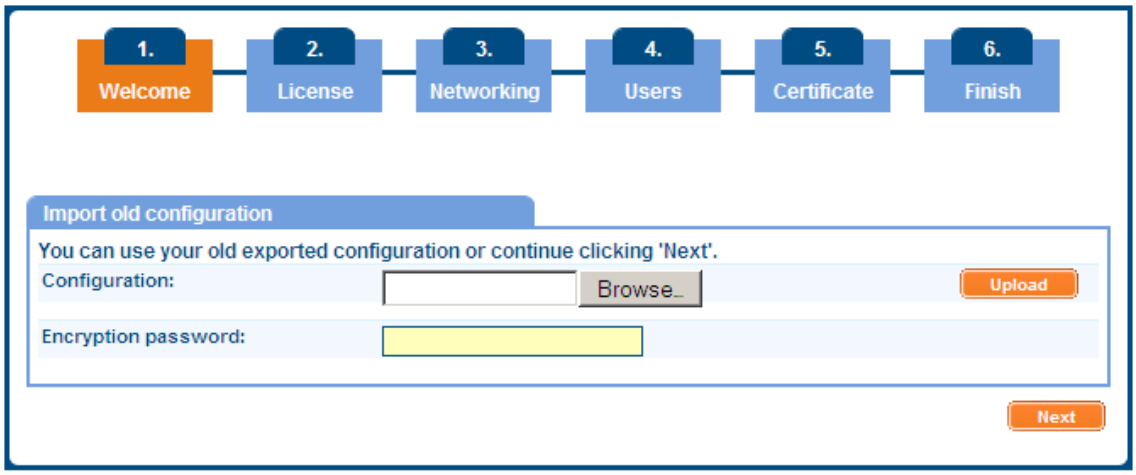


Figure 3.6. The Welcome Wizard

It is also possible to import an existing configuration from a backup file. Use this feature to restore a backup configuration after a recovery, or to migrate an existing SCB configuration to a new device.

Step a. Click **Browse** and select the configuration file to import.

Note
It is not possible to directly import a GPG-encrypted configuration into SCB, it has to be decrypted locally first.

Step b. Enter the passphrase used when the configuration was exported into the **Encryption passphrase** field.
For details on restoring configuration from a configuration backup, see *Procedure 11.8, Restoring configuration from a configuration backup (p. 245)*

Step c. Click **Import**.

Warning
If you use the Import function to copy a configuration from one SCB to another, do not forget to configure the IP addresses of the second SCB. Having two devices with identical IP addresses on the same network leads to errors.

Step 3. Accept the End User License Agreement and install the SCB license

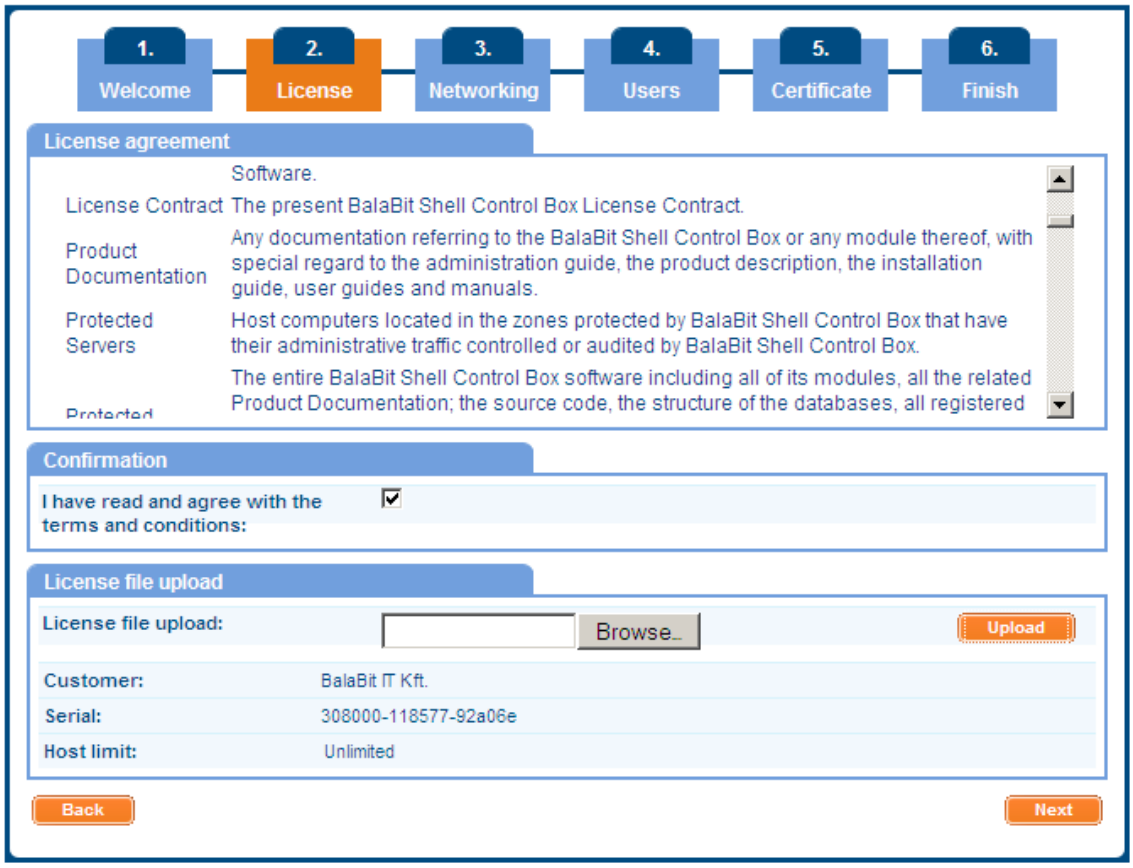

1. Welcome
2. License
3. Networking
4. Users
5. Certificate
6. Finish
License agreement
Software.
License Contract
The present BalaBit Shell Control Box License Contract.
Product Documentation
Any documentation referring to the BalaBit Shell Control Box or any module thereof, with special regard to the administration guide, the product description, the installation guide, user guides and manuals.
Protected Servers
Host computers located in the zones protected by BalaBit Shell Control Box that have their administrative traffic controlled or audited by BalaBit Shell Control Box.
The entire BalaBit Shell Control Box software including all of its modules, all the related Product Documentation; the source code, the structure of the databases, all registered
Confirmation
I have read and agree with the terms and conditions:
License file upload
License file upload:
Browse...
Upload
Customer:
BalaBit IT Kft.
Serial:
308000-118577-92a06e
Host limit:
Unlimited
Back
Next

Figure 3.7. The EULA and the license key

Step a. Read the End User License Agreement and select **Accept**.

Step b. Click **Browse**, select the SCB license file received with SCB, then click **Upload**.

Step c. Click **Next**.

Step 4. Fill the fields to configure networking. The meaning of each field is described below. The background of unfilled required fields is red. All parameters can later be modified using the regular interface of SCB.

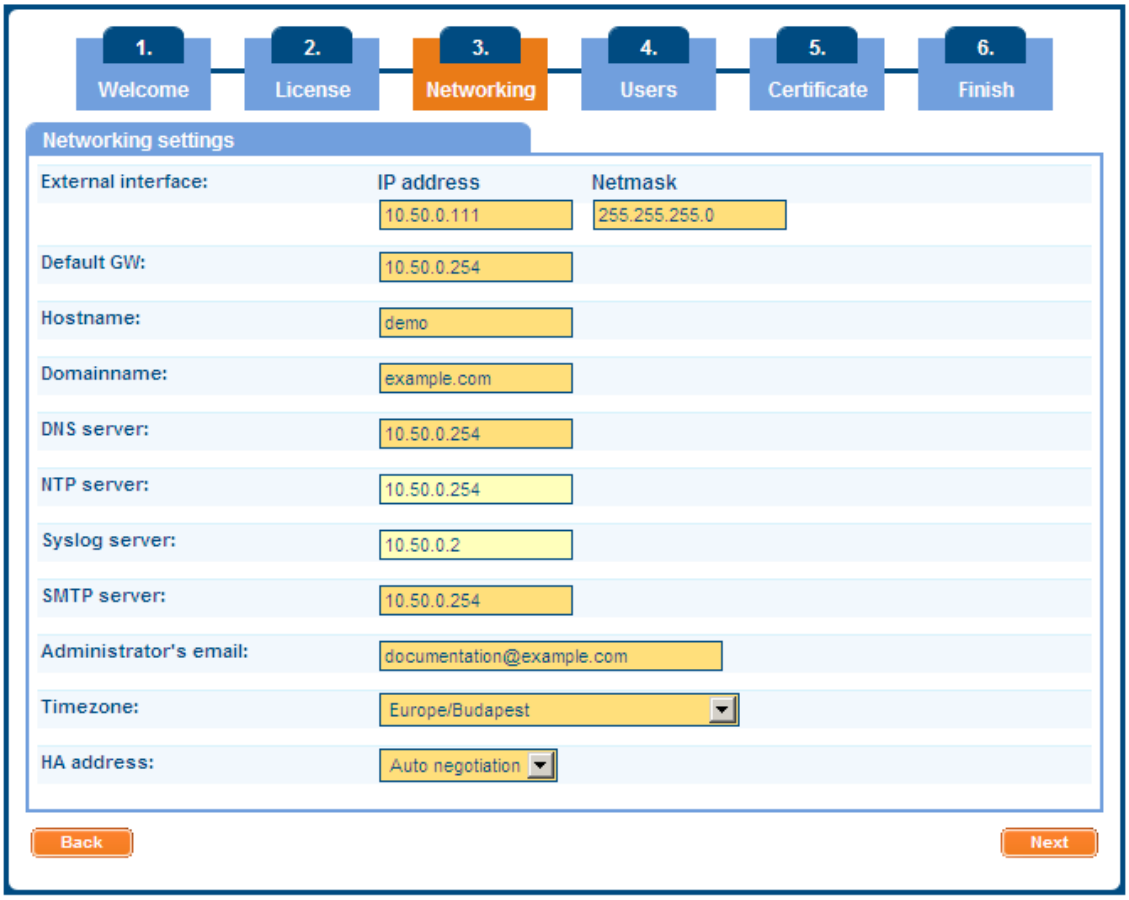


Figure 3.8. Initial networking configuration

Step a. *Hostname*: Name of the machine running SCB (for example `SCB`).

Step b. *Domain name*: Name of the domain used on the network.

Step c. *DNS server*: IP address of the name server used for domain name resolution.

Step d. *NTP server*: The IP address or the hostname of the NTP server.

Step e. *Syslog server*: The IP address or the hostname of the syslog server.

Step f. *SMTP server*: The IP address or the hostname of the SMTP server used to deliver e-mails.

Step g. *Administrator's e-mail*: E-mail address of the SCB administrator.

Step h. *Timezone*: The timezone where the SCB is located.

Step i. *External interface — IP address*: IP address of the external interface of SCB (for example `192.168.1.1`). The IP address can be chosen from the range of the corresponding physical subnet. Clients will connect the external interface, therefore it must be accessible to them.

Note
Do not use IP addresses that fall into the following ranges:

- `1.2.0.0/16` (reserved for communication between SCB cluster nodes)
- `127.0.0.0/8` (localhost IP addresses)

Step j. *External interface — Netmask*: The IP netmask of the given range in IP format. For example, general class C networks have the 255.255.255.0 netmask.

Step k. When configuring SCB for **Router** mode, enter the IP address and Netmask for the internal interface as well. SCB connects to the servers via the internal interface.

Step l. *Default gateway*: IP address of the default gateway. When using several network cards, the default gateway is usually in the direction of the external interface.

Step m. *HA address*: The IP address of the high availability (HA) interface. Leave this field on `auto` unless specifically requested by the support team.

Step n. Click **Next**.

Step 5. Enter the passwords used to access SCB.

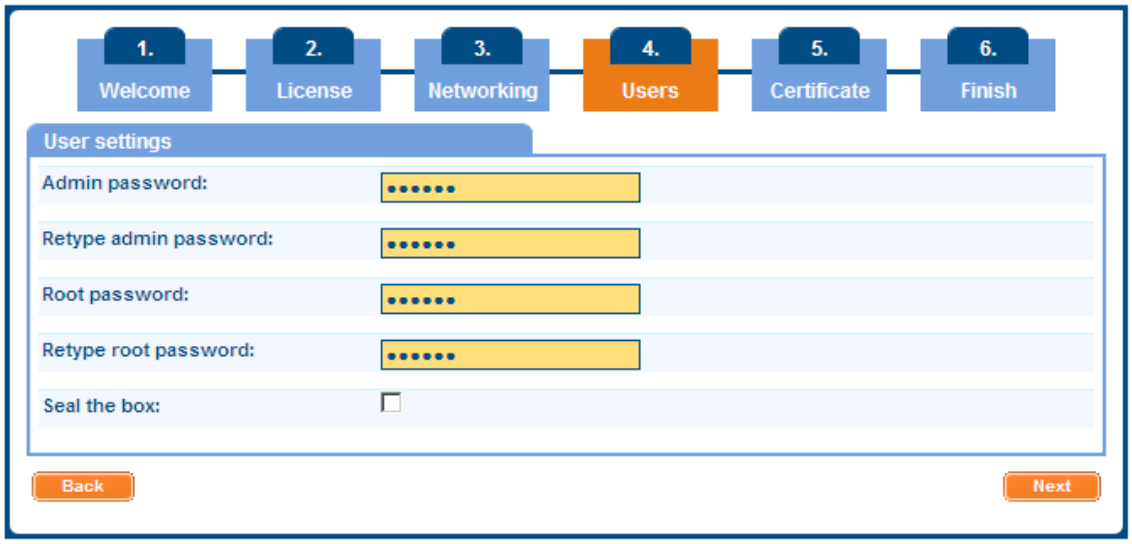


Figure 3.9. Passwords

Note
SCB passwords can contain the following special characters: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

Step a. *Admin password*: The password of the `admin` user who can access the web interface of SCB.

Step b. *Root password*: The password of the `root` user, required to access SCB via SSH or from the local console.

Note
Accessing SCB using SSH is rarely needed, and recommended only for advanced users for troubleshooting situations.

Step c. If you want to prevent users from accessing SCB remotely via SSH or changing the root password of SCB, select the **Seal the box** checkbox. Sealed mode can be activated later from the web interface as well. For details, see *Section 4.5.5, Sealed mode (p. 113)*.

Step d. Click **Next**.

Step 6. Upload or create a certificate for the SCB web interface. This SSL certificate will be displayed by SCB to authenticate administrative HTTPS connections to the web interface.

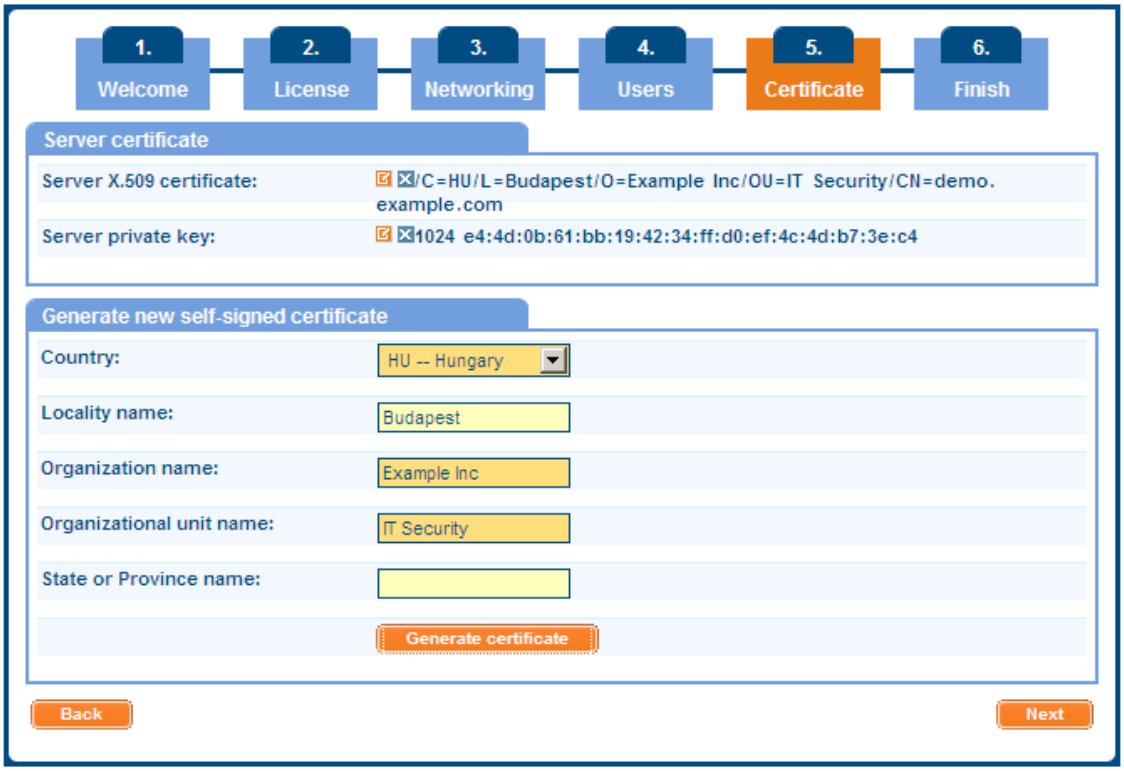


Figure 3.10. Creating a certificate for SCB

To create a self-signed certificate, fill the fields of the **Generate new self-signed certificate** section and click **Generate**. The certificate will be self-signed by the SCB appliance; the hostname of SCB will be used as the issuer and common name.

Step a. *Country*: Select the country where SCB is located (for example HU-Hungary).

Step b. *Locality*: The city where SCB is located (for example Budapest).

Step c. *Organization*: The company who owns SCB (for example Example Inc.).

Step d. *Organization unit*: The division of the company who owns SCB (for example IT Security Department).

Step e. *State or Province*: The state or province where SCB is located.

Step f. Click **Generate**.

If you want to use a certificate that is signed by an external Certificate Authority, in the **Server X.509 certificate** field, click to upload the certificate.

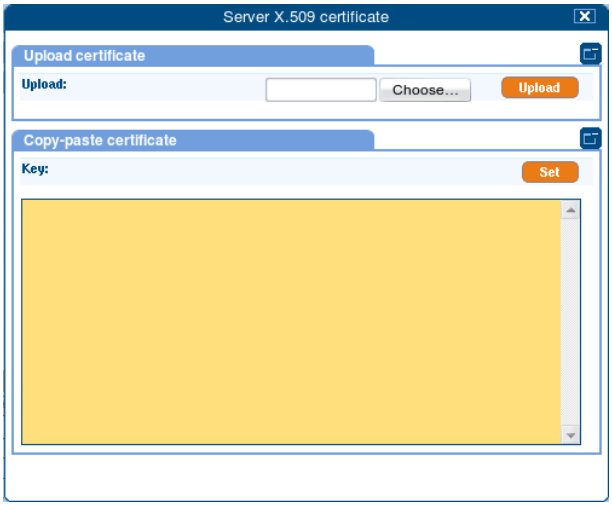


Figure 3.11. Uploading a certificate for SCB

Then in the **Server private key** field click , upload the private key, and enter the password protecting the private key.

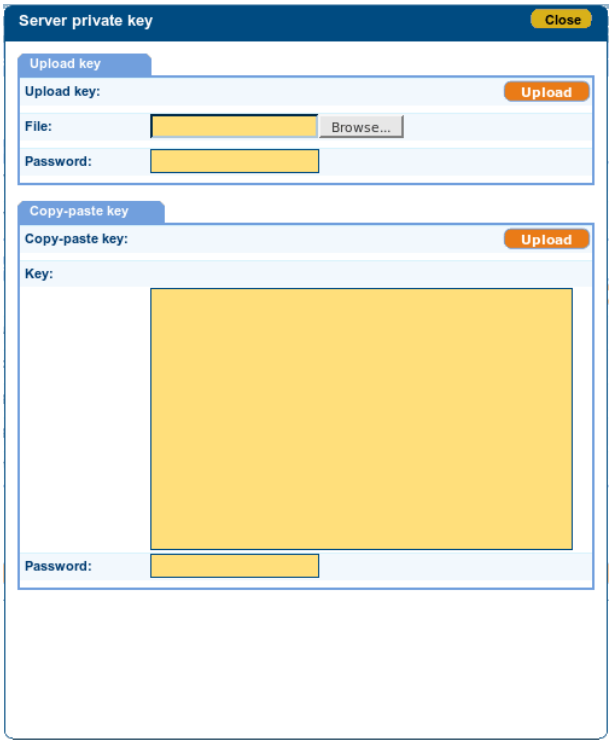


Figure 3.12. Uploading a private key

Note
SCB accepts private keys in PEM (RSA and DSA), PUTTY, and SSHCOM/Tectia format. Password-protected private keys are also supported.

Step 7. Review the data entered in the previous steps. This page also displays the certificate generated in the last step; the RSA SSH key of SCB, and information about the license file.

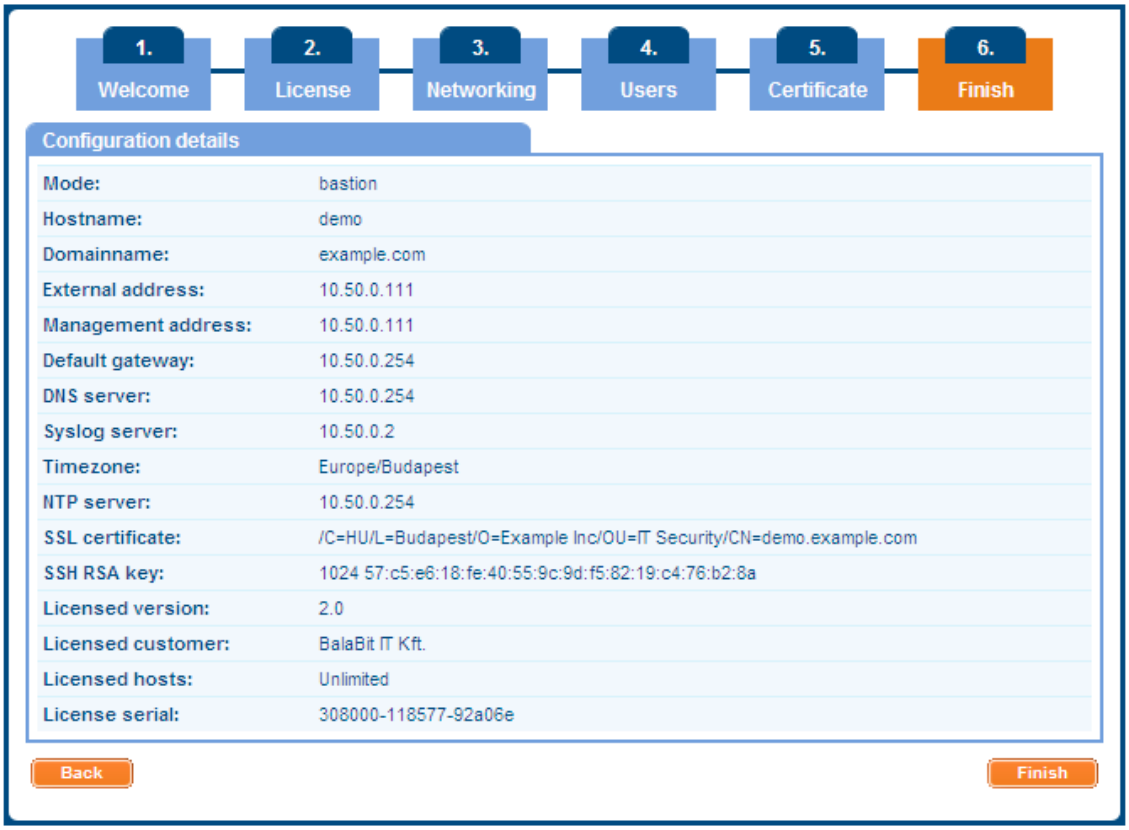


Figure 3.13. Review configuration data

If all information is correct, click **Finish**.

Warning
The configuration takes effect immediately after clicking **Finish**. Incorrect network configuration data can render SCB unaccessible.

SCB is now accessible from the regular web interface via the IP address of its external interface.

Step 8.

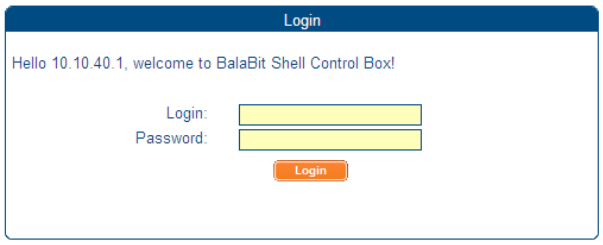


Figure 3.14. Logging in to SCB

Your browser is automatically redirected to the IP address set as the external interface of SCB, where you can login to the web interface of SCB using the `admin` username and the password you set for this user in the Welcome Wizard.

3.3. Procedure – Logging in to SCB and configuring the first connection

Purpose:

After finishing the initial configuration of SCB using the Welcome Wizard, connections must be configured between the clients and the servers. SCB inspects only the connections that are configured from the web interface; all other connections are forwarded without any inspection. The procedure below describes how to enable a simple SSH terminal or a Remote Desktop session.

Steps:

Step 1. Login to SCB's web interface.

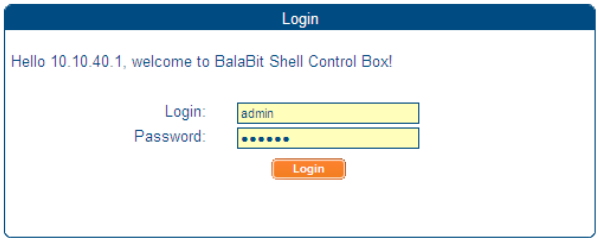


Figure 3.15. The first login

Step a. Open the `https://IP-address-of-the-external-interface/` page from your browser to access the web interface of SCB. Replace the `IP-address-of-the-external-interface` string with the IP set for the external interface in *Configuring the external interface (p. 33)* of the Welcome Wizard (for example `192.168.1.1`).

Step b. The certificate created in *Creating the web interface certificate (p. 34)* of the Welcome Wizard is displayed. Accept it.

Step c. Login to the SCB web interface using the displayed login screen.

- Enter `admin` into the **Login** field.
- Enter the password set in *Setting the administrator password (p. 33)* for the `admin` user into the **Password** field.

- Click **Login**. The main page of the SCB administration interface is displayed.

Step 2. Configure a new connection. This connection will allow any user to connect to a specified server from a specific client machine.

Step a.
- To configure an SSH connection, select **SSH Control > Connections** from the Main Menu. Only terminal sessions will be permitted.
- To configure an RDP connection, click on the **RDP Control > Connections** from the Main Menu. Only basic Remote Desktop sessions will be permitted (that is, no clipboard or file-sharing).

Step b. Click on the + icon on the left to create a new connection.

Step c. Enter a name into the **Name** field that will identify the connection (for example `admin-mainserver`).

Tip
It is recommended to use descriptive names that give information about the connection, for example refer to the name of the accessible server, the allowed users, and so on.

Step d. *Bastion mode*: Enter the IP addresses defining the connection. Skip this step if SCB is set to Router or Bridge mode.

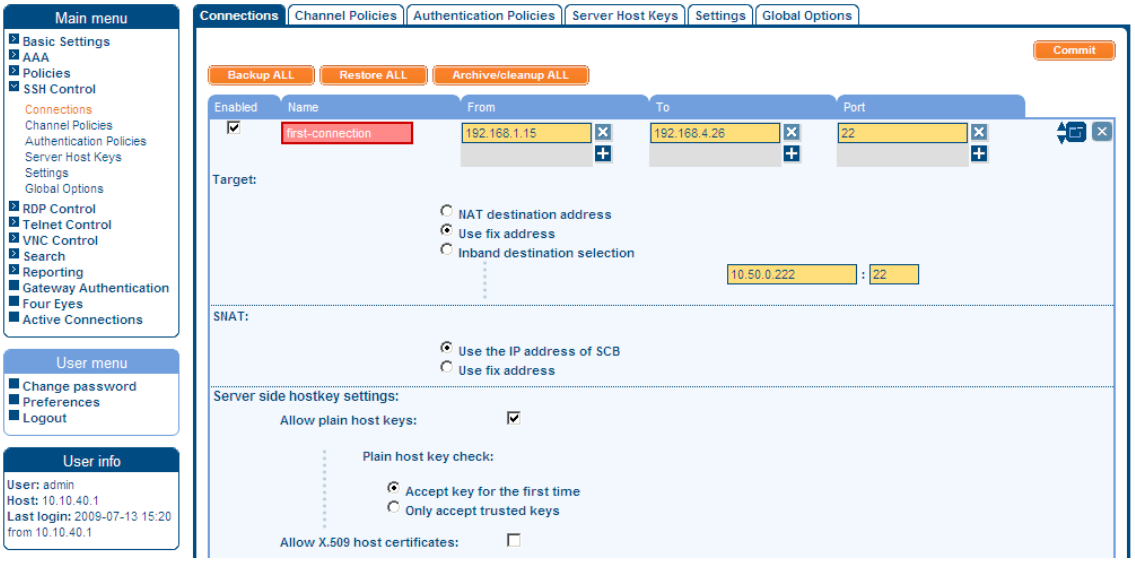


Figure 3.16. A connection in Bastion mode

- Enter the IP address of the client that will be permitted to access the server into the **From** field.
- Enter the IP address of SCB's external interface into the **To** field.
- Enter a port number into the **Port** field.

- Enter the IP address of the server into the **Use fix address** field of the **Target** section.
- Enter the port number where the server is accepting connections into the **Port** field of the **Target** section.

Step e. *Router mode* and *Bridge mode*: Enter the IP addresses defining the connection. Skip this step if SCB is set to Bastion mode.

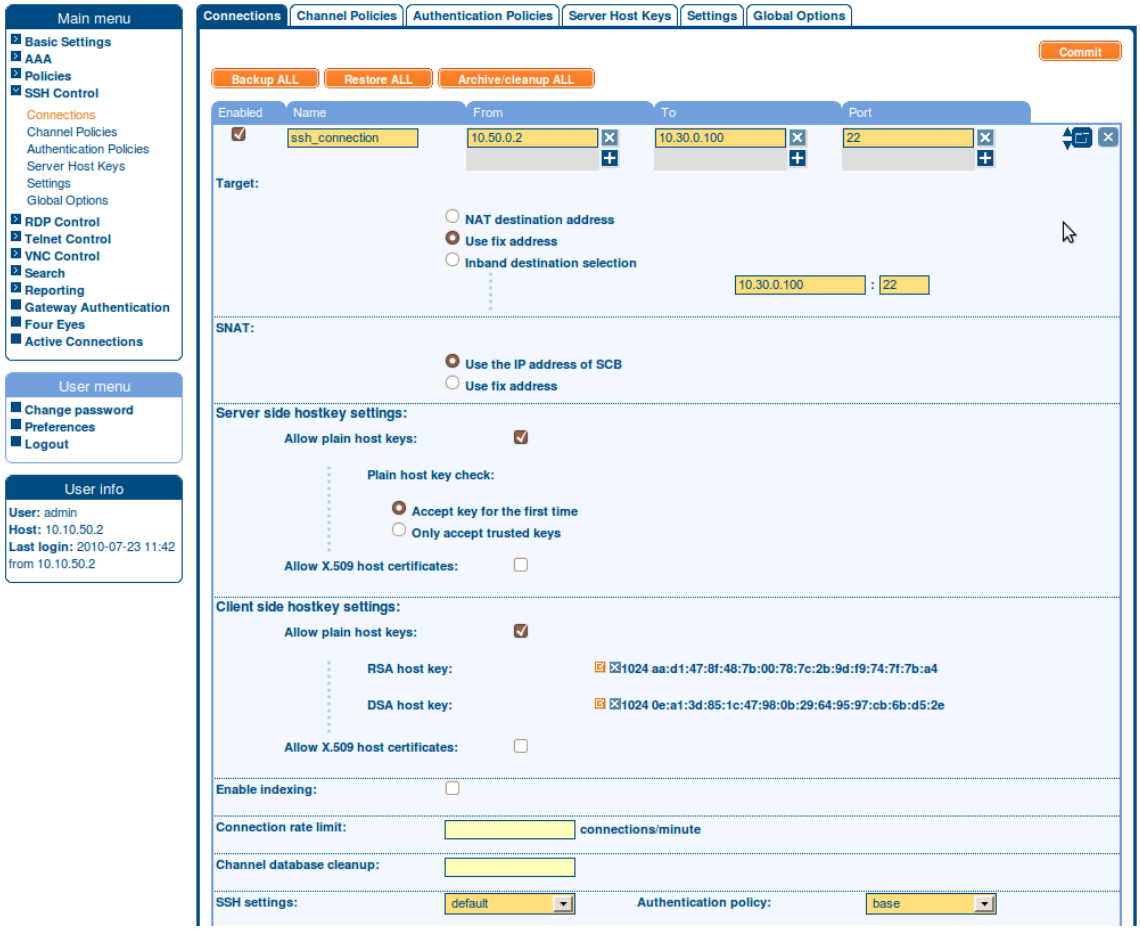


Figure 3.17. A connection in Router mode

- Enter the IP address of the client that will be permitted to access the server into the **From** field.
- Enter the IP address of the server into the **To** field.
- Enter the port number where the server is accepting connections into the **Port** field.

Step f. Click **Commit**.
This connection allows any user from the client machine to connect to the specified server, but permits only terminal sessions — other SSH channels like TCP forwarding are disabled.

Step 3. Test the new configuration: try to initiate an SSH or and RDP connection from the client to the server. Note that in Bastion mode the client addresses SCB's external IP address and the port set in *Configuring a connection in Bastion mode (p. 38)*, while IP address of the the server set in *Configuring a connection in Router mode (p. 39)* in Router mode.

Chapter 4. Configuring and managing SCB

SCB is configured via the web interface. Configuration changes take effect automatically after clicking Commit. Only the modifications of the current page or tab are activated — each page and tab must be committed separately.

This chapter describes how to configure, manage, and maintain SCB.

- For details about the supported browsers, see *Section 4.1, Supported web browsers and operating systems (p. 41)*.
- For details on how to use the web interface of SCB, see *Section 4.2, The structure of the web interface (p. 41)*.
- For details on how to configure the basic settings of the SCB host like network settings, system monitoring, and backup settings, see *Section 4.3, Basic settings (p. 45)*.
- For details on how to manage user accounts and privileges see *Section 4.4, User management and access control (p. 75)*.
- For details on how to control, reboot, update, and troubleshoot SCB, see *Section 4.5, Managing SCB (p. 91)*.

4.1. Supported web browsers and operating systems

Supported browsers: Mozilla Firefox 3.0 or newer and Microsoft Internet Explorer 7 and 8. The browser must support HTTPS connections, JavaScript, and cookies. Make sure that both JavaScript and cookies are enabled.

Note
SCB displays a warning message if your browser is not supported or JavaScript is disabled.

Supported operating systems: Microsoft Windows XP, Windows 2003 Server, Windows Vista, Windows 2008 Server, Windows 7, and Linux.

The SCB web interface can be accessed only using SSLv3 or TLSv1 encryption and strong cipher algorithms.

Note that when using Internet Explorer 7 on Windows 2008 to access SCB you must enable active scripting for the Internet Zone, otherwise the SCB web interface will not operate properly. This is not required on other platforms or browser versions. To accomplish this, select **Tools > Internet Options > Security > Internet Zone > Custom level**, and set the **Active scripting** field to `Enabled`.

When using Internet Explorer 8.0 on Windows Server 2008 R2 Enterprise, disable the Content Advisor in the Internet Explorer. To accomplish this, select **Tools > Internet Options > Content > Content Advisor > Disable**.

4.2. The structure of the web interface

The web interface consists of the following main sections:

Main menu: Each menu item displays its options in the main workspace on one or more tabs. Click ▾ in front of a main menu item to display the list of available tabs.

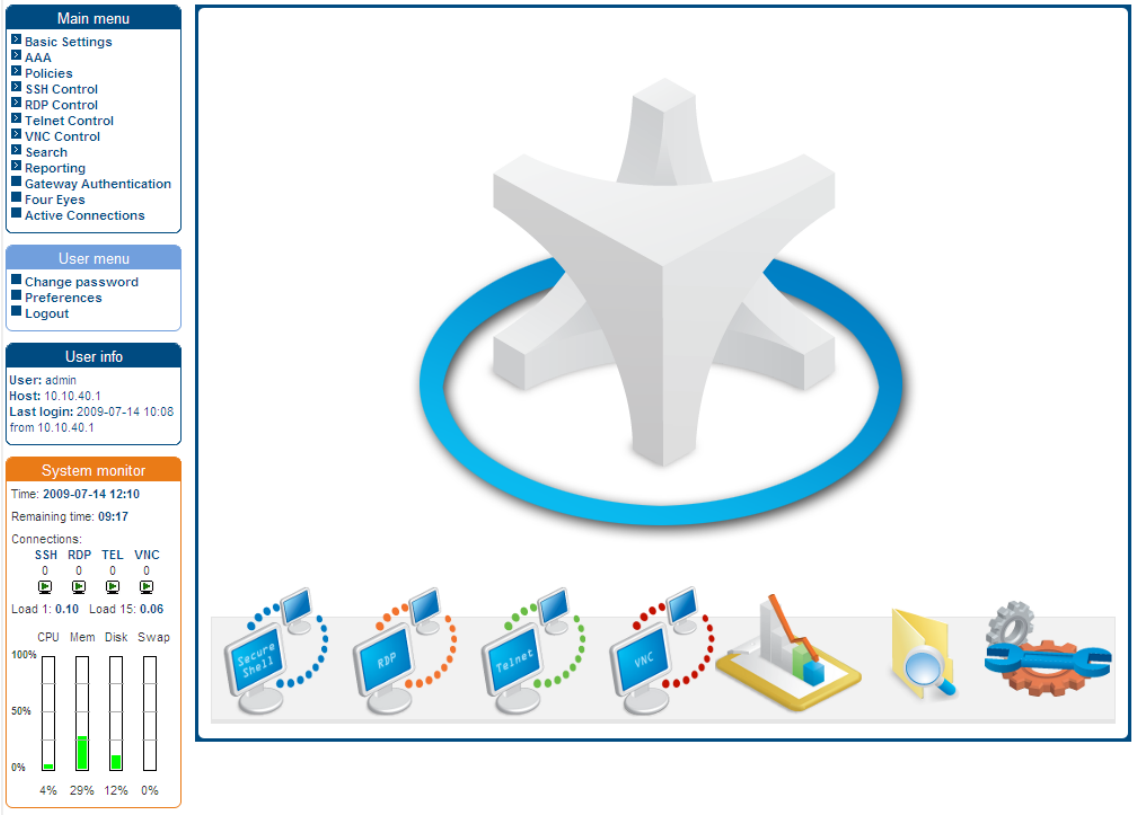


Figure 4.1. Structure of the web interface

User menu: Provides possibilities to change your SCB password; to log out; and disable confirmation dialogs and tooltips using the **Preferences** option.

User info: Provides information about the user currently logged in:

- **User:** username
- **Host:** IP address of the user's computer
- **Last login:** date and IP address of the user's last login

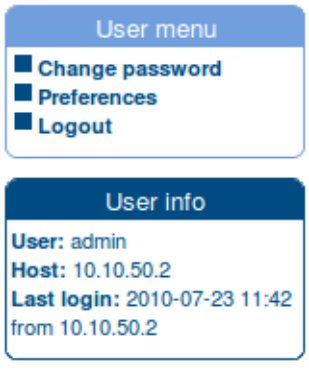


Figure 4.2. User menu and user info

System monitor: Displays accessibility and system health information about SCB, including the following:

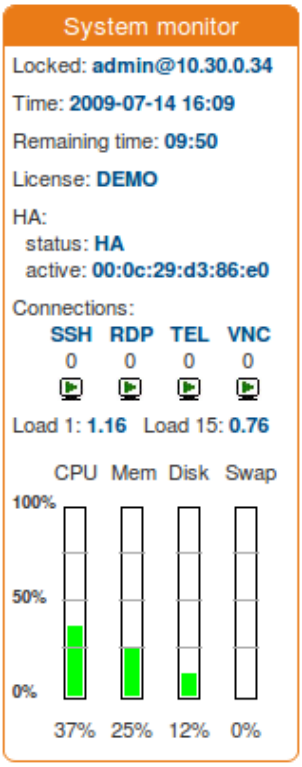


Figure 4.3. System monitor

- **Time:** System date and time.
- **Remaining time:** The time remaining before the session to the web interface times out.
- Indicators if SSH, RDP, Telnet, and VNC traffic is permitted to the protected servers.
- The number of active SSH, RDP, Telnet, and VNC connections.
- License information if the license is not valid, or an evaluation version license has expired.
- The status of the RAID devices, if synchronization between the disks is in progress.
- **HA:**The HA status and the ID of the active node if two SCB units are running in a High Availability cluster. If there are redundant Heartbeat interfaces configured, their status is displayed as well. If the nodes of the cluster are synchronizing data between each other, the progress and the time remaining from the synchronization process is also displayed.
- Average system load during the
  - **Load 1:** last minute
  - **Load 15:** last fifteen minutes
- CPU, memory, hard disk, and swap use. Hover the mouse above the graphical bars to receive a more details in a tooltip, or navigate to **Basic Settings > Dashboard** for detailed reports.

The System monitor displays current information about the state of SCB. To display a history of these parameters, go to **Basic Settings > Dashboard**. For details, see *Section 11.6, Status history and statistics (p. 240)*.

4.2.1. Elements of the main workspace

The main workspace displays the configuration settings related to the selected main menu item grouped into one or more tabs. Related parameters of a tab are organized into labeled groups or sections, marked with blue outline Interfaces.

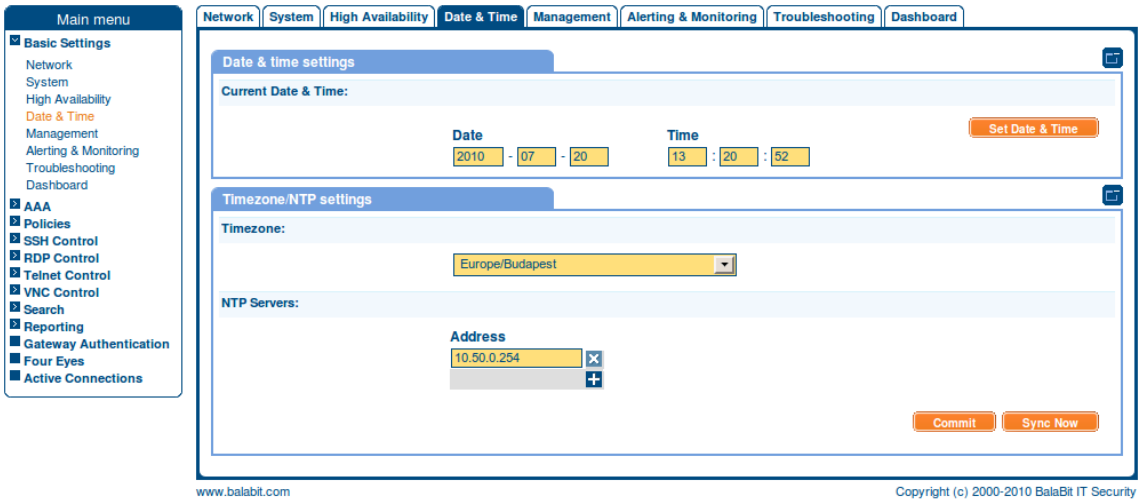


Figure 4.4. Main workspace

- Commit Each page includes one or more orange action buttons. The most common action button is the **Commit**, which saves and activates the changes of the page.
- / *Show/Hide Details*: Displays or hides additional configuration settings and options.
- , *Create entry*: Create a new row or entry (for example an IP address or a policy).
- , *Delete entry*: Delete a row or an entry (for example an IP address or a policy).
- , *Open/collapse lists*: Open or close a list of options (for example the list of available reports).
- *Modify entries or upload files*: Edit an entry (for example a host key, a list, and so on), or upload a file (for example a private key). These actions open a popup window where the actual modification can be performed.
- , *Position an item in a list*: Modify the order of items in a list. The order of items in a list (for example the order of connections, permitted channels in a channel policy, and so on) is important because when SCB is looking for a policy, it evaluates the list from top to down, and selects the first item completely matching the search criteria. For example, when a client initiates a connection to a protected server, SCB selects the first connection policy matching the client's IP address, the server's IP address, and the target port (the From, To, and Port fields of the connection).

Message window: This popup window displays the responses of SCB to the user's actions, for example *Configuration saved successfully*. Error messages are also displayed here. All messages are included in the system log. For detailed system logs (including message history), see the **Troubleshooting** tab of the Basic menu. To make the window

appear only for failed actions, navigate to **User menu > Preferences** and enable the **Autoclose successful commit messages** option.

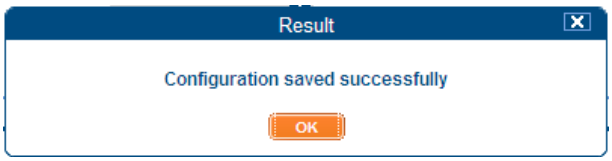


Figure 4.5. Message window.

4.2.2. Multiple web users and locking

Multiple administrators can access the SCB web interface simultaneously, but only one of them can modify the configuration. This means that the configuration of SCB is automatically locked when the first administrator who can modify the configuration accesses a configuration page (for example the **Basic Settings**, **AAA**, or **Logs** menu). The username and IP address of the administrator locking the configuration is displayed in the **System Monitor** field. Other administrators must wait until the locking administrator logs out, or the session of the administrator times out. However, it is possible to access the **Search** and **Reporting** menus,and to perform gateway authentication and 4-eyes authorization or browse the configuration with only View rights (for details, see *Section 4.4.6, Managing user rights and usergroups (p. 84)*).

Note
If an administrator logs in to SCB using the local console or a remote SSH connection, access via the web interface is completely blocked. Inactive local and SSH connections timeout just like web connections. For details, see *Section 4.5.4, Accessing the SCB console (p. 110)*.

4.3. Basic settings

Use the **Basic Settings** option of the main menu to configure the SCB host. The exact options are described in the following sections.

- For details on how to configure the network interfaces, name resolution, and other networking-related settings, see *Section 4.3.1, Network settings (p. 46)*.
- For details on how to control (for example reboot) SCB, upload a new firmware or license, export the current configuration, or stop the incoming syslog traffic, see *Section 4.3.1, Network settings (p. 46)*.
- For details on how to set the system time and automatic time synchronization to an NTP server, see *Section 4.3.2, Date and time configuration (p. 50)*.
- For details on how to configure where SNMP and e-mail alerts are sent, see *Section 4.3.3, System logging, SNMP and e-mail alerts (p. 52)*.
- For details on how to configure system monitoring and alerts, see *Section 4.3.4, Configuring system monitoring on SCB (p. 57)*.
- For details on how to configure data archiving and backups, see *Section 4.3.5, Data and configuration archiving and backups (p. 62)*.

4.3.1. Network settings

The **Network** tab contains the network interface and naming settings of SCB.

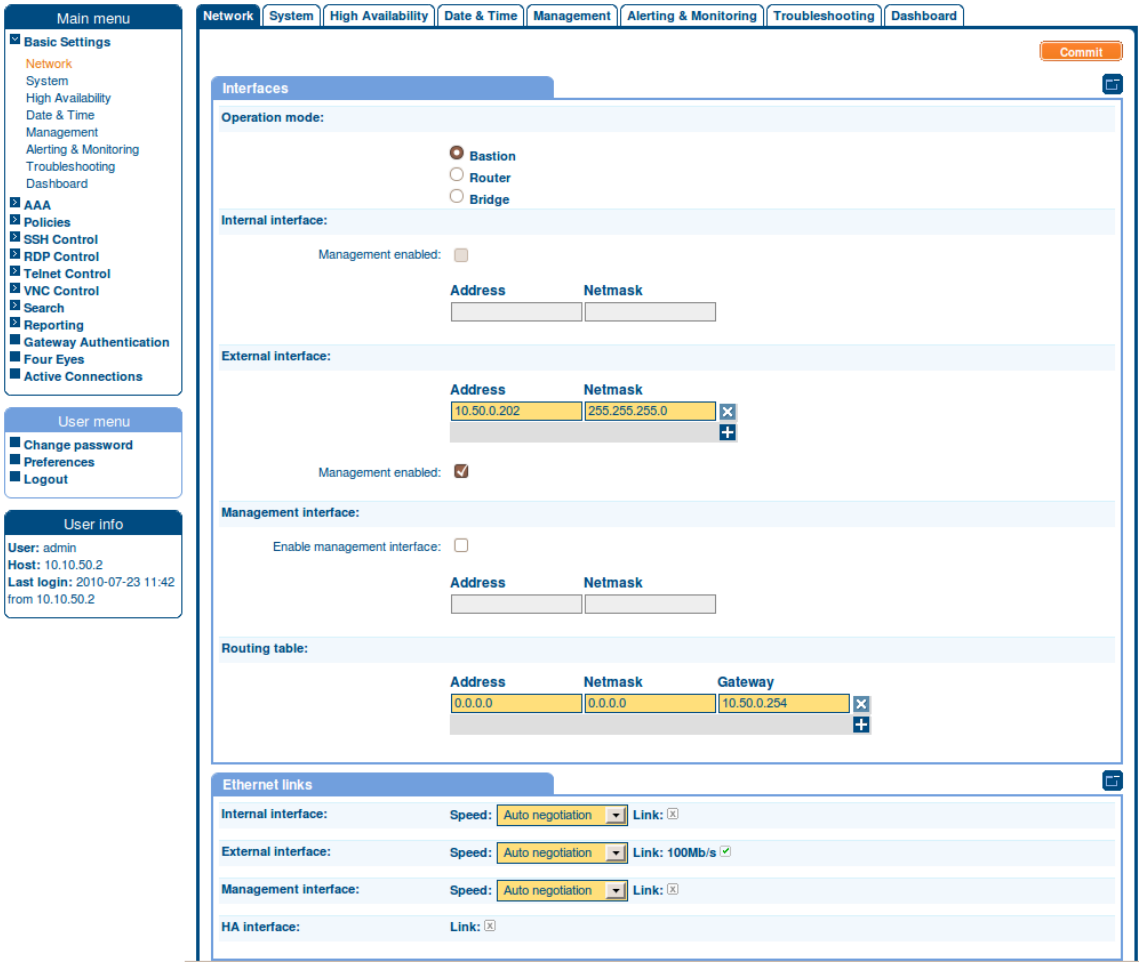


Figure 4.6. Network settings

- *Operation mode*: SCB can operate in Bastion, Router, or Bridge mode. For details, see *Section 2.3, Modes of operation (p. 6)*.

- *Internal interface*: The IP address and Netmask of the SCB network interface that connects to the protected servers. The internal interface is available only in Router mode.

Note
Do not use IP addresses that fall into the following ranges:

- `1.2.0.0/16` (reserved for communication between SCB cluster nodes)
- `127.0.0.0/8` (localhost IP addresses)

To enable access to the SCB web interface from the internal interface (even if the management interface is configured), select the **Enable management** option.

- *External interface*: The Address and Netmask of the SCB network interface that receives client connections. Click the + and x icons to add new alias IP addresses (also called alias interfaces) or delete existing ones. At least one external interface must be configured. If the management interface is disabled, the SCB web interface can be accessed via the external interface. When multiple external interfaces are configured, the first one refers to the physical network interface, all others are alias interfaces. The SCB web interface can be accessed from all external interfaces (if no management interface is configured).

  Optionally, you can enable access to the SCB web interface even if the management interface is configured by activating the **Management enabled** function.

Warning
If you enable management access on an interface and configure alias IP address(es) on the same interface, SCB will accept management connections only on the original address of the interface.

Note
Do not use IP addresses that fall into the following ranges:

- `1.2.0.0/16` (reserved for communication between SCB cluster nodes)
- `127.0.0.0/8` (localhost IP addresses)

Note
The speed of the interface is displayed for every interface. To explicitly set the speed of the interface, select the desired speed from the **Speed** field. Modifying the speed of the interface is recommended only for advanced users. Also note that changing the interface speed might not take effect if the network card of SCB has been replaced with one different from the original.

- *Management interface*: The Address and Netmask of the SCB network interface used to access the SCB web interface. If the management interface is configured, the web interface can be accessed only via this interface, unless access from other interfaces is explicitly enabled.

Note
Do not use IP addresses that fall into the following ranges:

- `1.2.0.0/16` (reserved for communication between SCB cluster nodes)
- `127.0.0.0/8` (localhost IP addresses)

4.3.1.1. Procedure – Configuring the management interface

Purpose:

To activate the interface, complete the following steps.

Steps:

Step 1. Navigate to **Basic Settings > Network > Interfaces**.

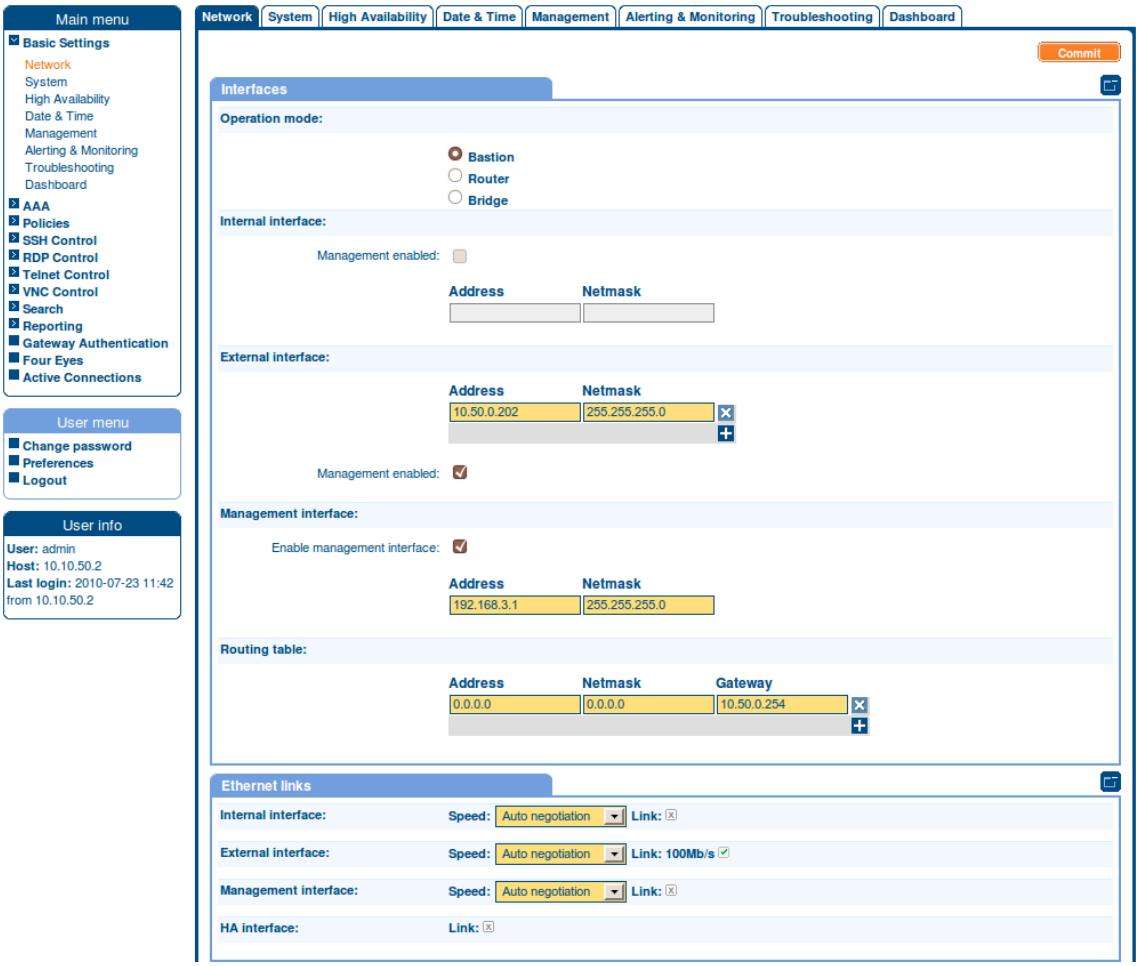


Figure 4.7. Configuring the management interface

Step 2. In the **Management interface** field, select **Enable management interface**.

Step 3. Into the **Address** field, enter the IP address of SCB's management interface.

Step 4. Into the **Netmask** field, enter the netmask related to the IP address.

Step 5.

Warning
After clicking **Commit**, the web interface will be available only via the management interface — it will not be accessible using the current (external) interface, unless the **Management enabled** option is selected for the external interface.

Ensure that the Ethernet cable is plugged and the management interface is connected to the network; this is indicated by a green check icon in the **Basic settings > Networks > Ethernet links > HA interface > Link** field. When using High Availability, ensure that the management interface of both SCB units is connected to the network.

The **HA interface** section indicates if a link is detected on the high availability interface.

Click **Commit**.

- *HA address*: The IP address of the high availability (HA) interface. Leave this field on `Auto negotiation` unless specifically requested by the support team.

Note
As of SCB version 2.0.2, when both nodes of a cluster boot up in parallel, the node with the `1.2.4.1` HA IP address will become the master node.

- *Interfaces* > *Routing table*: When sending a packet to a remote network, SCB consults the routing table to determine the path it should be sent. If there is no information in the routing table then the packet is sent to the default gateway. Use the routing table to define static routes to specific hosts or networks. You have to use the routing table if the internal interface is connected to multiple subnets, because the default gateway is (usually) towards the external interface. Click the ⊞ and ⊠ icons to add new routes or delete existing ones. A route means that messages sent to the *Address*/*Netmask* network should be delivered to *Gateway*.

  For detailed examples, see *Procedure 4.3.1.2, Routing management traffic to the management interface (p. 49).*

- *Naming* > *Hostname*: Name of the machine running SCB.

- *Naming* > *DNS search domain*: Name of the domain used on the network. When resolving the domain names of the audited connections, SCB will use this domain to resolve the target hostname if the append domain entry is of a target address is empty.

- *Naming* > *Primary DNS server*: IP address of the name server used for domain name resolution.

- *Naming* > *Secondary DNS server*: IP address of the name server used for domain name resolution if the primary server is unaccessible.

4.3.1.2. Procedure – Routing management traffic to the management interface

Purpose:

For security reasons — and also to reduce network usage on the external and internal interfaces — it is recommended to direct all management-related traffic of SCB towards the management network interface. Such traffic includes access to the web interface, *backups and archiving*, *syslog messages* , and *e-mail or SNMP alerts* sent to the administrator.

Warning
Complete the following procedure only if the management interface is configured; otherwise the data sent by SCB will be lost. For details on configuring the management interface, see *Procedure 4.3.1.1, Configuring the management interface (p. 47).*

Steps:

Step 1. To add a new routing entry, navigate to **Basic Settings > Network > Interfaces** and in the **Routing table** field, click ⊞.

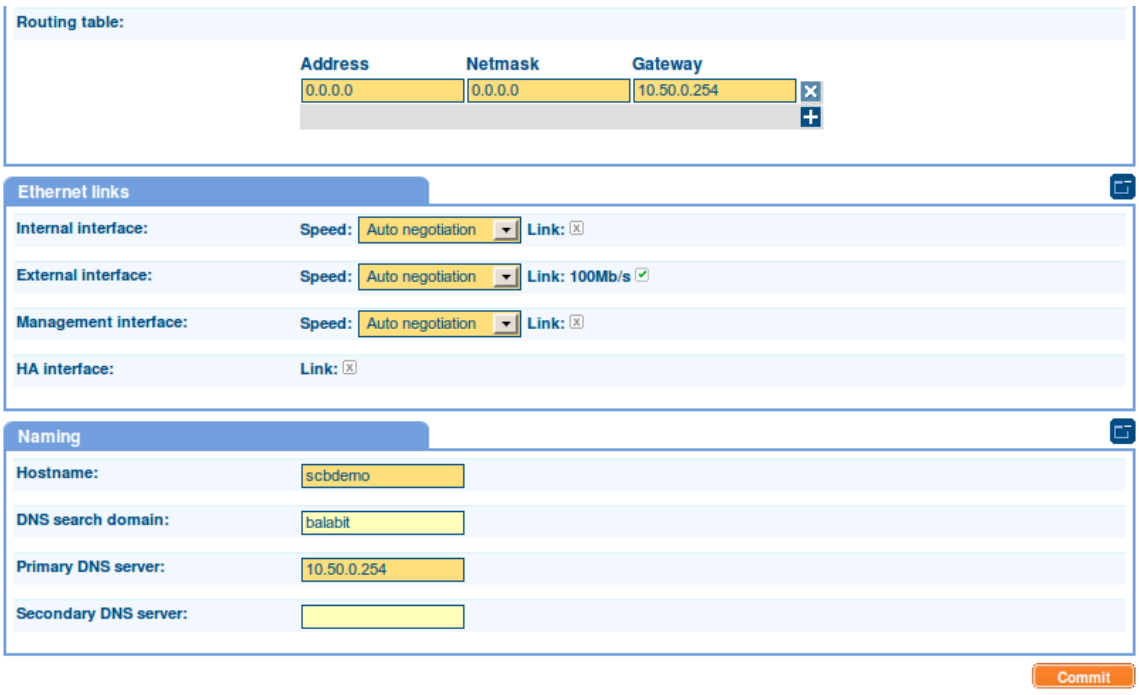


Figure 4.8. Routing

Step 2. Enter the IP address of the backup server (as set in *Procedure 4.3.5.1, Creating configuration and data backups (p. 62)*) into the **Address** field.

Step 3. Enter the related netmask into the **Netmask** field.

Step 4. Enter the IP address of the gateway used on that subnetwork into the **Gateway** field.

Step 5. Click **Commit**.

Step 6. Repeat Steps 1-5 and create a routing entry for other backup servers if needed.

Step 7. Repeat Steps 1-5 and create a routing entry for the syslog server (as set in *Section 4.3.3, System logging, SNMP and e-mail alerts (p. 52)*).

Step 8. Repeat Steps 1-5 and create a routing entry for the SMTP server (as set in *Section 4.3.3, System logging, SNMP and e-mail alerts (p. 52)*).

4.3.2. Date and time configuration

Date and time related settings of SCB can be configured on the **Date & Time** tab of the **Basic** page.

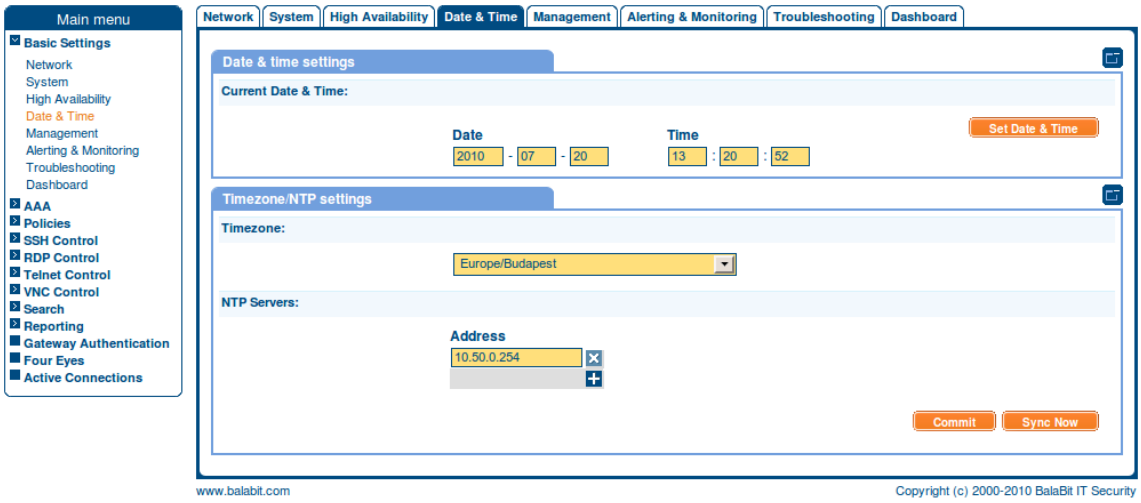


Figure 4.9. Date and time management

Warning
It is essential to set the date and time correctly on SCB, otherwise the date information of the logs and audit trails will be inaccurate.

SCB displays a warning on this page and sends an alert if the time becomes out of sync.

To explicitly set the date and time on SCB, enter the current date into respective fields of the **Date & Time Settings** group and click **Set Date & Time**.

4.3.2.1. Procedure – Configuring a time (NTP) server

Purpose:

To retrieve the date automatically from a time server, complete the following steps.

Steps:

Step 1. Select your timezone in the **Timezone** field.

Step 2. Enter the IP address of an NTP time server into the **Address** field.

Step 3. Click **Commit**.

Step 4. Click the + and ⊠ icons to add new servers or delete existing ones.

Note
If the time setting of SCB is very inaccurate (that is, the difference between the system time and the actual time is great), it might take a long time to retrieve the date from the NTP server. In this case, click **Sync now** to sync the time immediately using SNTP.

When two SCB units are operating in high availability mode, the **Sync now** button is named **Sync Master**, and synchronizes the time of the master node to the NTP server. To synchronize the time between the master and the slave nodes, click **Sync Slave to Master**.

4.3.3. System logging, SNMP and e-mail alerts

E-mail alerts and system logging can be configured on the **Basic Settings > Management** page.

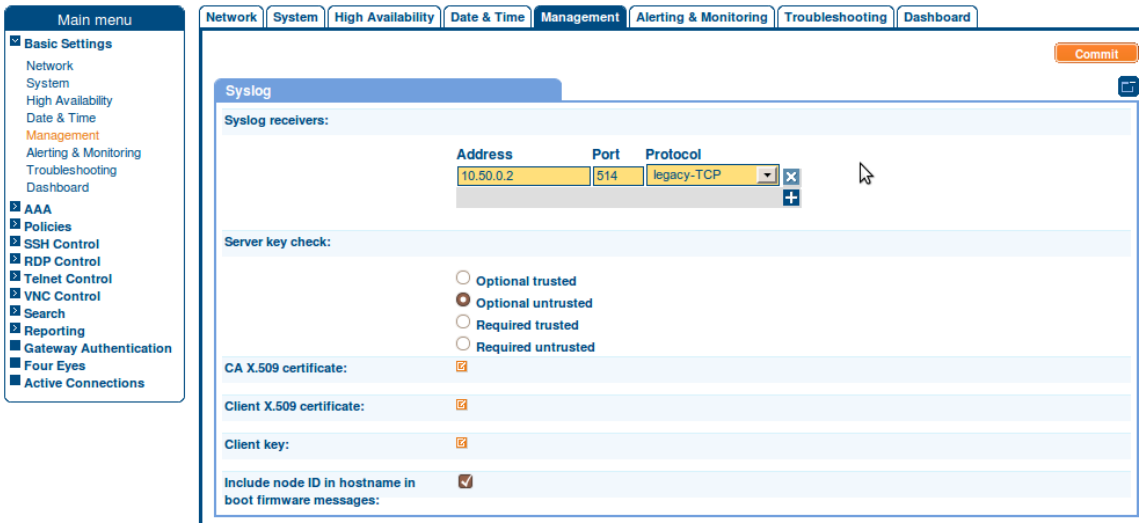


Figure 4.10. Configuring system logging and e-mail alerts

4.3.3.1. Procedure – Configuring system logging

Purpose:

SCB can send its system log messages to remote syslog servers. To configure logging to a remote server, complete the following steps:

Steps:

Step 1. Navigate to **Basic Settings > Management**.

Step 2. Click ➕ in the **Syslog receivers** field to add a new syslog server.

Step 3. Enter the IP address and port of the syslog server into the respective fields.

Step 4. Select the network protocol used to transfer the messages in the **Protocol** field. The `legacy-` prefix corresponds to the legacy BSD-syslog protocol described in RFC3164, while the `ietf-` prefix corresponds to the new IETF-syslog protocol described in RFC5424. Note that not every syslog server supports the IETF protocol yet.
Select `TCP+TLS` to send the log messages using a TLS-encrypted connection.

Tip
Transferring the syslog messages using TCP ensures that the server receives them.

Transferring the syslog messages using TLS encryption ensures that third parties cannot read the messages. However, not every syslog server accepts encrypted connections. The syslog-ng Premium Edition and Open Source Edition applications, and the syslog-ng Store Box (which is a log-collector appliance similar to SCB) support

both encrypted connections and the new IETF-syslog protocol as well. For details on these products, see *syslog-ng Premium Edition* and *syslog-ng Store Box*.

Step 5. To display separate hostnames for syslog messages sent by the nodes of a SCB HA cluster, select the **Include node ID in hostname in boot firmware messages** option. The node ID included in the hostname filed of the syslog message is the MAC address of the node's HA interface. (Messages of the core firmware are always sent by the master node.)

Step 6. If you have selected the `TCP+TLS` protocol, complete the following steps. Otherwise, click **Commit**.

Step a. If you want SCB to verify the certificate of the syslog server, select **Required trusted** in the **Server key check** field and proceed to the next step.

If you want SCB to simply accept any certificate shown by the server, select **Optional untrusted** in the **Server key check** field.

Note
Alternatively, you can use the following, less strict options to check the certificate of the server:

- *Optional trusted*: If the server sends a certificate, SCB checks if it is valid (not expired) and that the Common Name of the certificate contains the domain name or the IP address of the server. If these checks fail, SCB rejects the connection. However, SCB accepts the connection if the server does not send a certificate.
- *Required untrusted*: SCB requests a certificate from the server, and rejects the connection if no certificate is received, if the certificate is not valid (expired), or if the Common Name of the certificate does not contain the domain name or the IP address of the server.

Step b. Click the icon in the **CA X.509 certificate** field. A popup window is displayed. Click **Browse**, select the certificate of the Certificate Authority (CA) that issued the certificate of the syslog server, then click **Upload**. Alternatively, you can paste the certificate into the **Copy-paste** field and click **Upload**.

SCB will use this CA certificate to verify the certificate of the server, and reject the connections if the verification fails.

Step c. If the syslog server requires mutual authentication, that is, it expects a certificate from SCB, generate and sign a certificate for SCB, then click the icon in the **Client X.509 certificate** field to upload the certificate. After that, click the icon in the **Client key** field and upload the private key corresponding to the certificate.

Step d. Click **Commit**.

Step 7. Click the and icons to add new servers or delete existing ones.

Note
To reduce the risk of the syslog server not receiving log messages from SCB because of a network outage or other problem with the syslog server, SCB buffers up to 10 Megabytes of log messages to its hard disk in case the syslog server becomes unaccessible.

4.3.3.2. Procedure – Configuring e-mail alerts

Purpose:

To configure e-mail alerts, complete the following steps:

Steps:

Step 1. Navigate to **Basic Settings > Management > Mail settings**.

Step 2. Enter the IP address or the hostname of the mail server into the **SMTP server address** field.

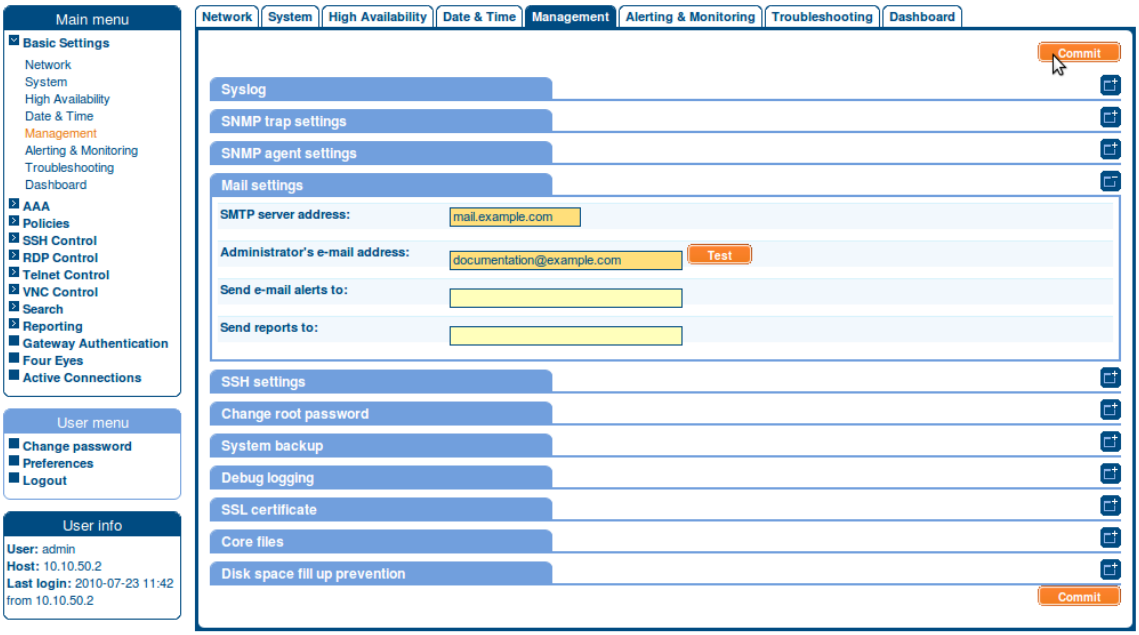


Figure 4.11. Configuring e-mail sending

Step 3. Enter the e-mail address of the administrator into the **Administrator's e-mail address** field. SCB sends notifications related to system-events (but not alerts and reports) to this address.

Step 4. Enter the e-mail address of the administrator into the **Send e-mail alerts to** field. SCB sends monitoring alerts to this address.

Step 5. Enter the e-mail address the person who should receive traffic reports from SCB into the **Send reports to** field. For details on reports, see *Section 6.3, Reports (p. 179)*.

Step 6. Click **Commit**.

Step 7. Click **Test** to send a test message.

If the test message does not arrive to the server, check if SCB can access the server. For details, see *Chapter 11, Troubleshooting SCB (p. 235)*.

Step 8. Navigate to **Basic Settings > Alerting & Monitoring** and select in which situations should SCB send an e-mail alert. For details, see *Section 4.3.4, Configuring system monitoring on SCB (p. 57)*.

Step 9. Click **Commit**.

4.3.3.3. Procedure – Configuring SNMP alerts

Purpose:

SCB can send alerts to a central monitoring server via SNMP (Simple Network Management Protocol). To configure SNMP alerts, complete the following steps:

Steps:

Step 1. Navigate to **Basic Settings > Management > SNMP trap settings**.

Step 2. Enter the IP address or the hostname of the SNMP server into the **SNMP server address** field.

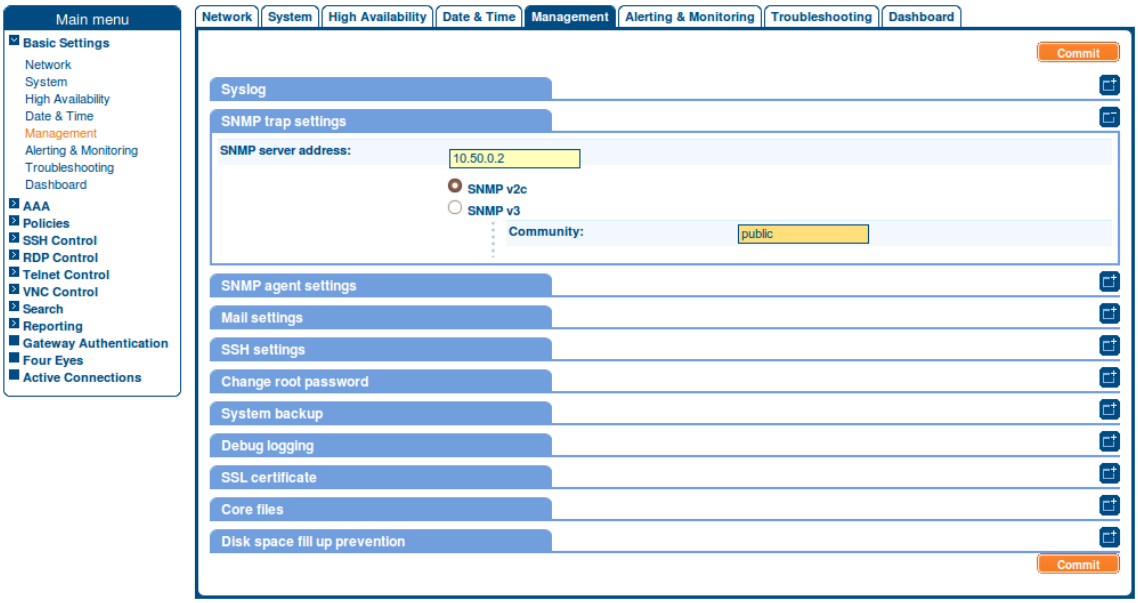


Figure 4.12. Configuring SNMP alerts

Step 3. Select the SNMP protocol to use.

- To use the SNMP v2c protocol for SNMP queries, select **SNMP v2c**, and enter the community to use into the **Community** field.
- To use the SNMP v3 protocol, select **SNMP v3** and complete the following steps:

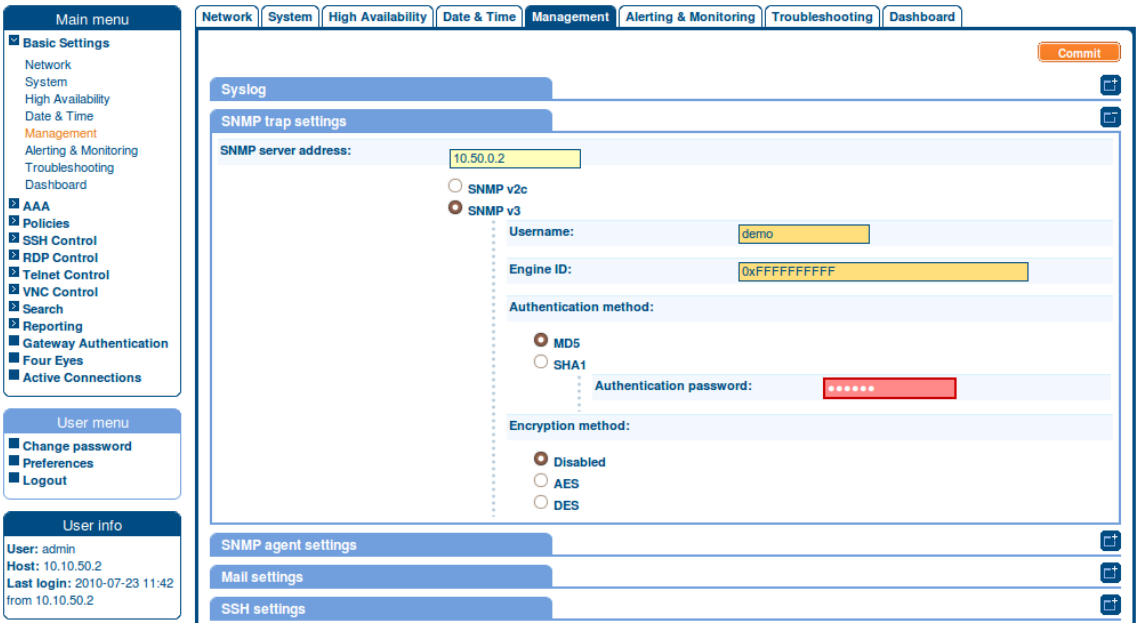


Figure 4.13. Configuring SNMP alerts using SNMPv3

Step a. Enter the username to use into the **Username** field.

Step b. Enter the engine ID to use into the **Engine ID** field. The engine ID is a hexadecimal number at least 10 digits long, starting with `0x`. For example `0xABABABABAB`.

Step c. Select the authentication method (`MD5`, `SHA1`) to use from the **Authentication method** field.

Step d. Enter the password to use into the **Authentication password** field.

Step e. Select the encryption method (`Disabled`, `DES`, `AES`) to use from the **Encryption method** field.

Step f. Enter the encryption password to use into the **Encryption password** field.

Step 4. Click **Commit**.

Step 5. Navigate to **Basic Settings > Alerting & Monitoring** and select in which situations should SCB send an SNMP alert. For details, see *Section 4.3.4, Configuring system monitoring on SCB (p. 57)*.

Step 6. Click **Commit**.

4.3.3.4. Procedure – Querying SCB status information using agents

Purpose:

External SNMP agents can query the status information of SCB. To configure which clients can query this information, complete the following steps:

Steps:

Step 1. Navigate to **Basic Settings > Management > SNMP agent settings**.

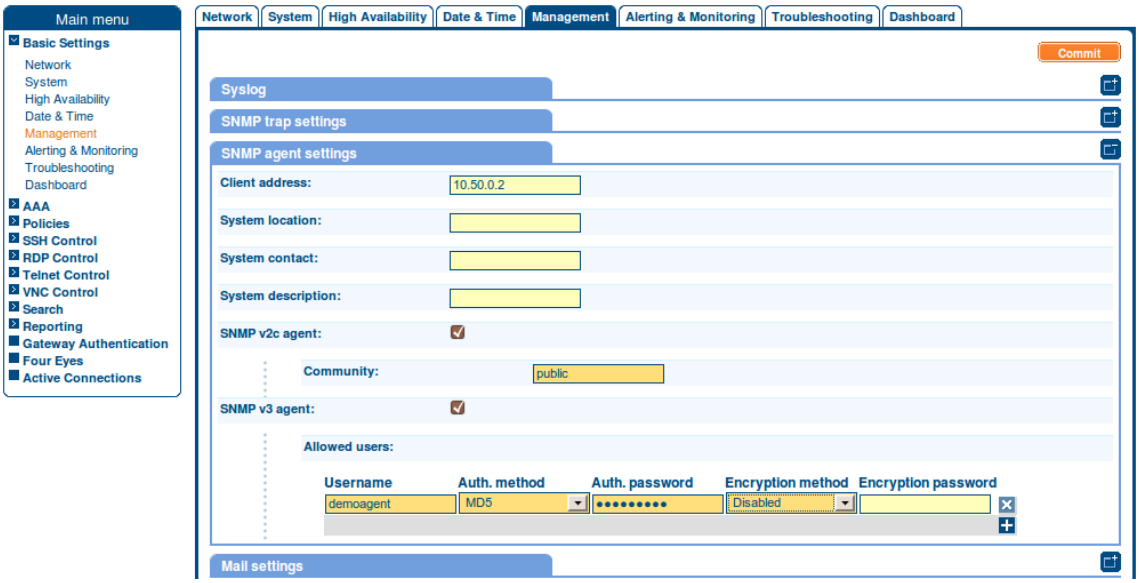


Figure 4.14. Configuring SNMP agent access

Step 2. The status of SCB can be queried dynamically via SNMP. By default, the status can be queried from any host. To restrict access to these data to a single host, enter the IP address of the host into the **Client address** field.

Step 3. Optionally, you can enter the details of the SNMP server into the **System location**, **System contact**, and **System description** fields.

Step 4. Select the SNMP protocol to use.

- To use the SNMP v2c protocol for SNMP queries, select **SNMP v2c agent**, and enter the community to use into the **Community** field.
- To use the SNMP v3 protocol, select **SNMP v3 agent** and complete the following steps:

  Step a. Click ⊞

  Step b. Enter the username used by the SNMP agent into the **Username** field.

  Step c. Select the authentication method (*MD5, SHA1*) to use from the **Auth. method** field.

  Step d. Enter the password used by the SNMP agent into the **Auth. password** field.

  Step e. Select the encryption method (*Disabled, DES, AES*) to use from the **Encryption method** field.

  Step f. Enter the encryption password to use into the **Encryption password** field.

  Step g. To add other agents, click ⊞.

Step 5. Click **Commit**.

4.3.4. Configuring system monitoring on SCB

SCB continuously monitors a number of parameters of the SCB hardware and its environment. If a parameter reaches a critical level (set in its respective **Maximum** field), SCB sends e-mail and SNMP messages to alert the administrator.

SCB sends SNMP alerts using the management network interface by default, or using the external interface if the management interface is disabled. SCB supports the SNMPv2c and SNMPv3 protocols. The SNMP server set on the **Management** tab can query status information from SCB.

Tip
In order have your central monitoring system to recognize the SNMP alerts sent by SCB, import the SCB-specific Management Information Base (MIB) into your monitoring system. Download both MIBs by navigating to **Basic Settings > Alerting & Monitoring** and clicking **Download MIBs** and import them into your monitoring system. For details, see the documentation of your monitoring system.

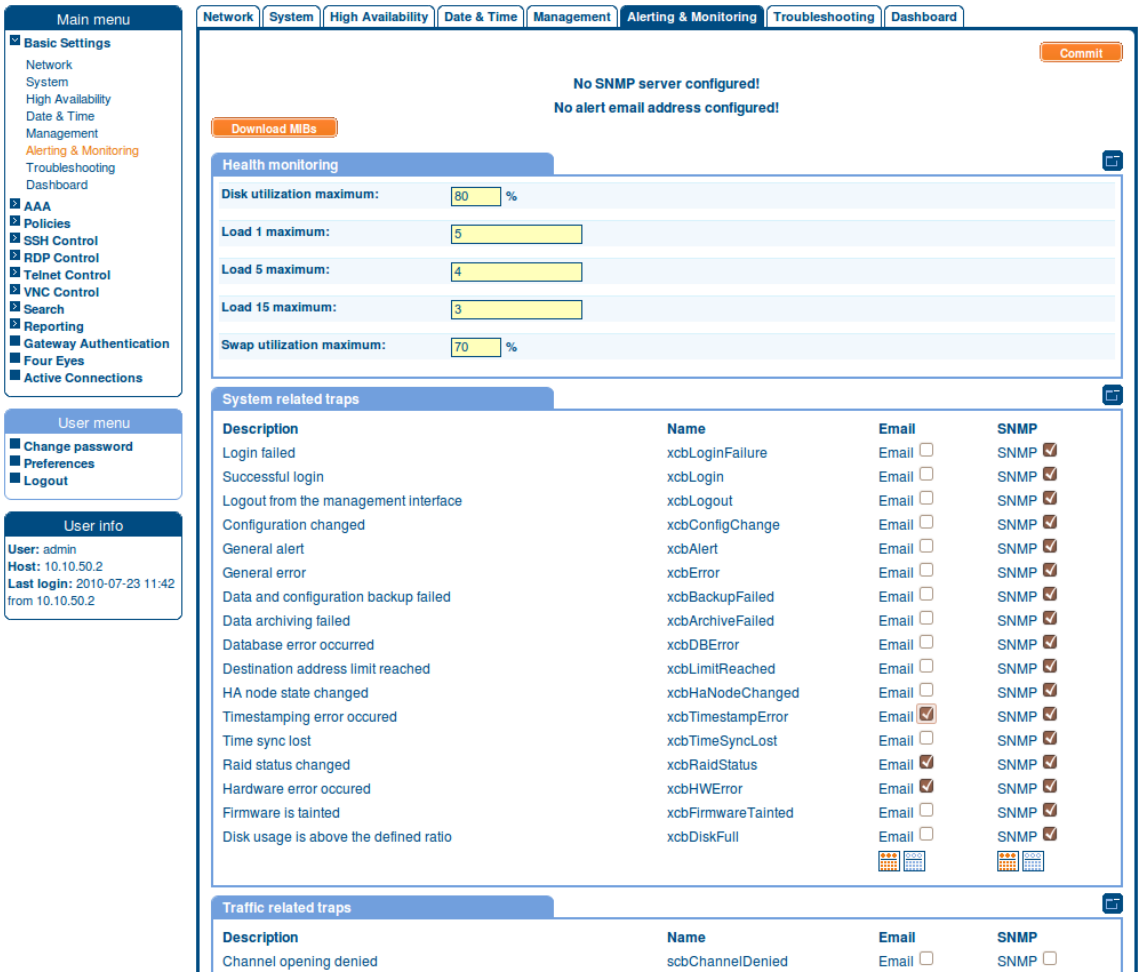


Figure 4.15. Configuring SNMP and e-mail alerting

4.3.4.1. Procedure – Configuring monitoring

Purpose:

To configure monitoring, complete the following steps:

Steps:

Step 1. Navigate to **Basic Settings > Alerting & Monitoring**.

Step 2. The default threshold values of the parameters are suitable for most situations. Adjust the thresholds only if needed.

Step 3. Click **Commit**.

Step 4. Navigate to **Basic Settings > Management** and verify that the **SNMP settings** and **Mail settings** of SCB are correct. SCB sends alerts only to the alert e-mail address and to the SNMP server.

Warning
Sending alerts fails if these settings are incorrect.

The following sections describe the parameters you can receive alerts on.

- For details on health-monitoring alerts, see *Section 4.3.4.2, Health monitoring (p. 59)*.
- For details on system-monitoring alerts, see *Section 4.3.4.4, System related traps (p. 60)*.
- For details on traffic-monitoring alerts, see *Section 4.3.4.5, Traffic-related alerts (p. 61)*.

4.3.4.2. Health monitoring

- *Disk utilization maximum*: Ratio of free space available on the hard disk. SCB sends an alert if the audit trails use more space than the set value. Archive the audit trails to a backup server to free disk space. For details, see *Procedure 4.3.5.2, Archiving the collected data (p. 66)*.

Note
The alert message includes the actual disk usage, not the limit set on the web interface. For example, you set SCB to alert if the disk usage increases above *10* percent. If the disk usage of SCB increases above this limit (for example to *17* percent), you receive the following alert message: *less than 90% free (= 17%)*. This means that the amount of used disk space increased above 10% (what you set as a limit, so it is less than 90%), namely to 17%.

- *Load 1|5|15 maximum*: The average load of SCB during the last one, five, or 15 minutes.
- *Swap utilization maximum*: Ratio of the swap space used by SCB. SCB sends an alert if it uses more swap space than the set value.

4.3.4.3. Procedure – Preventing disk space fill up

Purpose:

To prevent disk space from filling up, complete the following steps:

Steps:

Step 1. Navigate to **Basic Settings > Management > Disk space fill up prevention**.

Step 2. Set the limit of maximum disk utilization in percents in the respective field. When disk space is used above the set limit, SCB disconnects all clients. Entering *0* turns the feature off. The default value is *0*.

Step 3. *Optional step*: Enable the **Automatically start archiving** option to automatically start all configured archiving/cleanup jobs when disk usage goes over the limit.

Note
If there is no archiving policy set, enabling this option will not trigger automatic archiving.

Step 4. Navigate to **Basic Settings > Alerting & Monitoring > System related traps** and enable alert **Disk usage is above the defined ratio**.

Step 5. Click **Commit**.

4.3.4.4. System related traps

Name	SNMP alert ID	Description
Login failed	`xcbLoginFailure`	Failed login attempts from SCB web interface.
Successful login	`xcbLogin`	Successful login attempts into SCB web interface.
Logout from the management interface	`xcbLogout`	Logouts from SCB web interface.
Configuration changed	`xcbConfigChange`	Any modification of SCB's configuration.
General alert	`xcbAlert`	General alerts and error messages occurring on SCB. Note, that alerts on general alerts and errors are sent whenever there is an alert or error level message in the SCB system log. These messages are very verbose and mainly useful only for debugging purposes. Enabling these alerts may result in multiple e-mails or SNMP traps sent about the same event.
General error	`xcbError`	
Data and configuration backup failed	`xcbBackupFailed`	Alerts if the backup procedure is unsuccessful.
Data archiving failed	`xcbArchiveFailed`	Alerts if the archiving procedure is unsuccessful.
Database error occurred	`xcbDBError`	An error occurred in the database where SCB stores the connection metadata. Contact our support team (see *Section 5, Contact and support information (p. xiv)* for contact information).
Destination address limit reached	`xcbLimitReached`	The number of protected servers reached the limit set in the SCB license. Clients cannot connect to new servers using SCB.

Name	SNMP alert ID	Description
HA node state changed	`xcbHaNodeChanged`	A node of the SCB cluster changed its state, for example, a takeover occurred.
Timestamping error occured	`xcbTimestampError`	An error occurred during the timestaming process, for example the timestamping server did not respond.
Time sync lost	`xcbTimeSyncLost`	The system time became out of sync.
Raid status changed	`xcbRaidStatus`	The status of the node's RAID device changed its state.
Hardware error occured	`xcbHWError`	SCB detected a hardware error.
Firmware is tainted	`xcbFirmwareTainted`	A user has locally modified a file from the console.
Disk usage is above the defined ratio	`xcbDiskFull`	Disk space is used above the limit set in **Disk space fill up prevention**.

Table 4.1. System related traps

4.3.4.5. Traffic-related alerts

Name	SNMP alert ID	Description
Channel opening denied	`scbChannelDenied`	A user attempted to open a channel not permitted by the channel policy.
Connection denied	`scbConnectionDenied`	A user attempted to connect a server not permitted in the connection policies.
User successfully authenticated	`scbAuthSuccess`	A user successfully authenticated on a protected server using an SSH connection.
User authentication failed	`scbAuthFailure`	A user failed to complete the authentication on a protected server using an SSH connection.
SSH host key mismatch	`scbSshHostKeyMismatch`	The SSH host key of a server did not match the key stored on SCB.
New SSH host key learned	`scbHostKeyLearned`	SCB learned a new SSH host key.
Connection timed out	`scbConnectionTimedout`	A connection to a protected server timed out.
Connection to the server failed	`scbConnectionFailed`	A connection to a protected server failed.
Audit trail rate limit exceeded	`scbAuditRateLimit`	The growth of an audit trail exceeded the rate limit set in the **Global Options** of the protocol. This may have been caused by an a deliberate attack.
Audit trail size limit exceeded	`scbAuditSizeLimit`	The size of an audit trail exceeded the file limit set in the **Global Options** of the protocol.
User successfully authenticated on the gateway	`scbGWAuthSuccess`	A user has successfully authenticated a connection on SCB as part of a gateway-authentication process.
User authentication failed on the gateway	`scbGWAuthFailure`	The gateway-authentication of a connection has failed.

Name	SNMP alert ID	Description
User mapping failed on the gateway	`scbUserMappingFailure`	A usermapping policy did not find a suitable mapping for the connection.

Table 4.2. Traffic-related alerts

4.3.5. Data and configuration archiving and backups

The BalaBit Shell Control Box can create automatic backups of its configuration and the stored audit-trails to a remote server. Backups and archiving is controlled using backup and archiving policies that define the protocol to use, the address of the backup server, and other parameters.

4.3.5.1. Procedure – Creating configuration and data backups

Purpose:

To configure automatic configuration backups, complete the following steps:

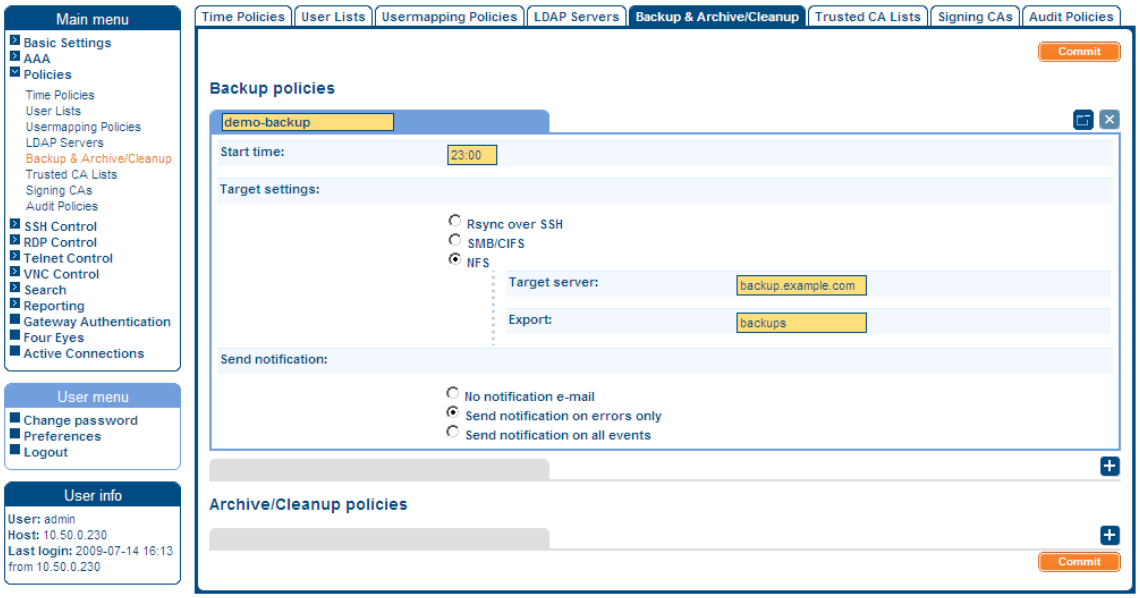


Figure 4.16. Configuring backups and archiving

Steps:

Step 1. Create a backup policy.

Step a. Navigate to **Policies > Backup & Archive/Cleanup** and click ⊞ in the **Backup policies** section to create a new backup policy.

Step b. Enter a name for the backup policy (for example `config-backup`).

Step c. Enter the time when the backup process should start into the **Start time** field in HH:MM format (for example `23:00`).

Step d. Enter the IP address or the hostname of the remote server into the **Target server** field (for example `backup.example.com`).

Step e.

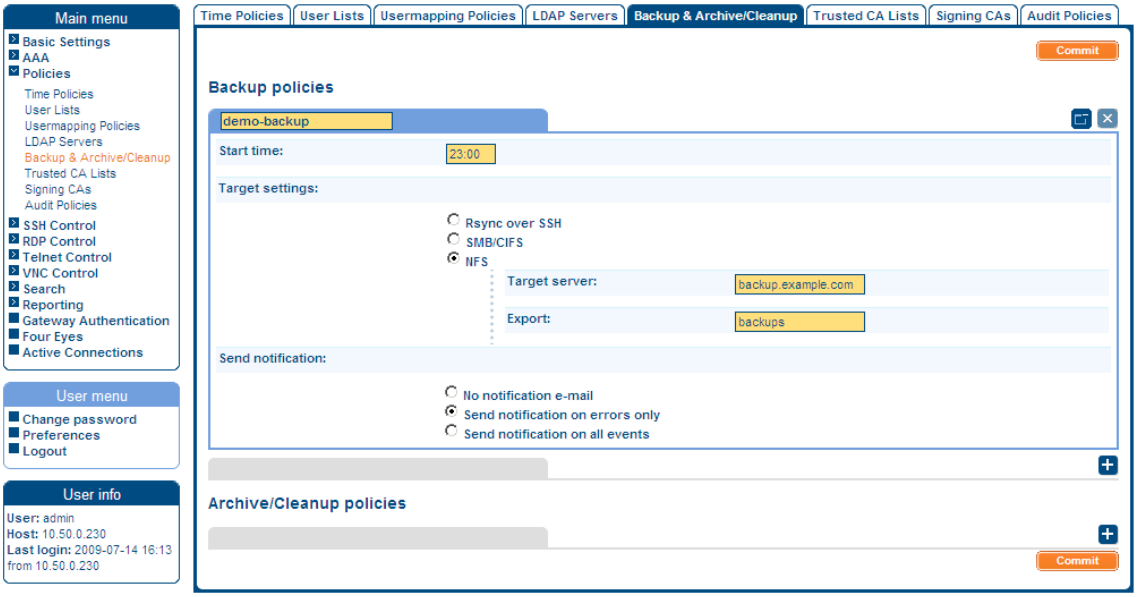


Figure 4.17. Configuring backups and archiving

SCB can access the remote server via different protocols. Select the one to use from the available protocols:

- *Rsync over SSH*: Execute the `rsync` command via the Secure Shell protocol. Note that the backup server must run rsync version 3.0 or newer.
- *SMB/CIFS*: Server Message Block protocol used on Microsoft Windows Network.

Warning
The CIFS implementation of NetApp storage devices is not compatible with the CIFS implementation used in SCB, therefore it is not possible to create backups and archives from SCB to NetApp devices using the CIFS protocol (the operation fails with a similar error message: *`/opt/scb/mnt/14719217504d41370514043/reports/2010": Permission denied (13) '2010/day/' rsync: failed to set times on`*).

To overcome this problem, either:

- use the NFS protocol to access your NetApp devices, or
- use a backup device that has a CIFS implementation compatible with SCB, for example, Windows or Linux Samba.

- *NFS*: Network File System protocol.

Warning
When using the NFS protocol to create backups or archives, ensure that the files on the remote server are readable for the *`www-data`* user as well, because SCB uses this user to access remote backups and archives if needed.

Step f. Provide the protocol-specific parameters for the selected method. The protocol-specific parameters are described in *Section 4.3.5.4, Parameters of the backup protocols (p. 70)*.

Step g. To receive e-mail notification of the backup, select the **Send notification on errors only** or the **Send notification on all events** option. Notifications are sent to the administrator e-mail address set on the **Management** tab, and include the list of the files that were backed up.

Warning
Starting with SCB 3.0, the notification e-mail does not include the list of backed up file. To include the list of files in the e-mail, select **Send notification on all events** and disable the **Omit file list** option. However, note that if list is very long (for example, SCB stores over 20000 audit trails), the SCB web interface might become unaccessible.

Note
This e-mail notification is different from the one set on the **Alerting & Monitoring** tab. This notification is sent to the administrator's e-mail address, while the alerts are sent to the alert e-mail address (see *Section 4.3.4, Configuring system monitoring on SCB (p. 57)*).

Step h. Click **Commit**.

Expected outcome:

A backup policy is created.

Step 2. To use this policy to create configuration backups, navigate to **Basic Settings > Management > System backup**, and select the backup policy you want to use for backing up the configuration of SCB in the **System backup policy** field.

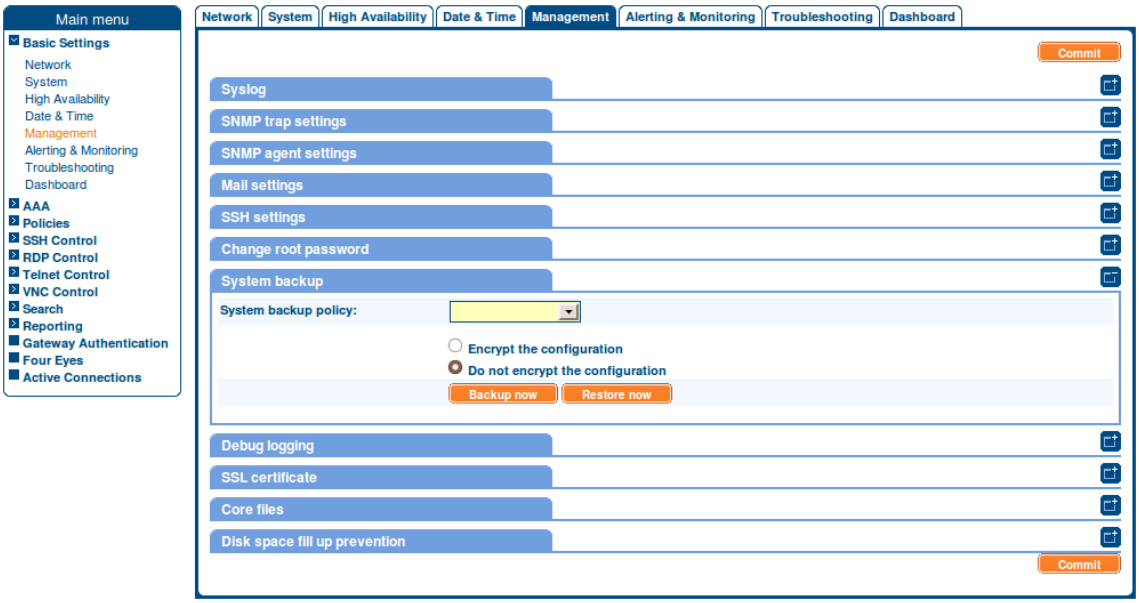


Figure 4.18. Configuring system backups

To use this policy to create data backups, navigate to **Connections**, select the connection you want to backup, and select a backup policy in the **Backup policy** field.

Step 3. Click **Commit**.

Tip
To create an immediate backup of SCB's configuration to your machine (not to the backup server), select **Basic Settings > System > Export configuration**.

Note that the configuration export contains only the system settings and configuration files (including changelogs). System backups includes additional information like reports and alerts, and also the connection database.

For details on restoring configuration from a configuration backup, see *Procedure 11.8, Restoring configuration from a configuration backup (p. 245)*

Note
Backup is different from archiving: the purpose of backup is to create a snapshot of SCB's configuration or the data stored on SCB that can be used for recovery in case of errors. Backup deletes all other data from the target directory; while restoring a backup deletes all other data from SCB.

Tip
To start the backup process immediately, click **Backup now**. The **Backup now** functionality works only after a backup policy has been selected.

To restore the stored data (audit trails, logs, reports, and so on), click **Restore now**. Note that the **Restore now** function does not restore the configuration files of SCB.

When restoring a data backup, you must restore the configuration backup (also called system backup) as well. Otherwise, SCB will not handle the restored data properly.

4.3.5.2. Procedure – Archiving the collected data

Purpose:

To configure data archiving, complete the following steps.

Steps:

Step 1. Create an archive policy.

Step a. Navigate to **Policies > Backup & Archive/Cleanup** and click ⊞ in the **Archive/Cleanup policies** section to create a new archive policy.

Step b.

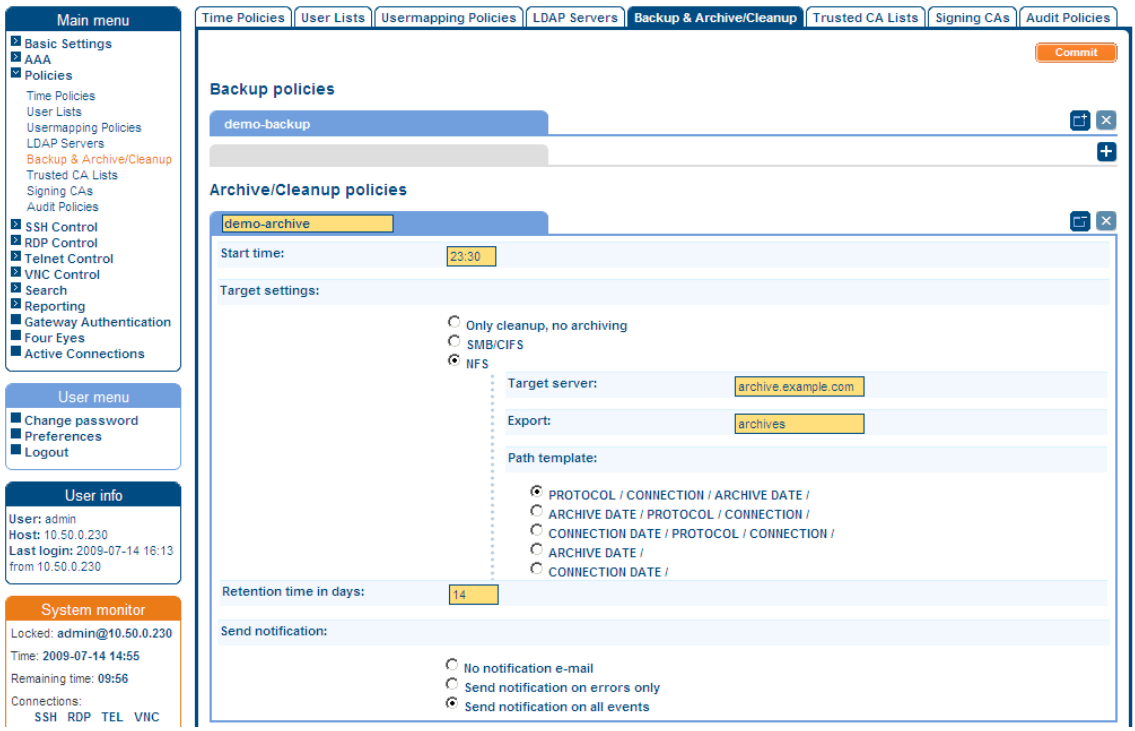


Figure 4.19. Configuring backups and archiving

Enter a name for the archive policy.

Step c. Enter the time when the archive process should start into the **Start time** field in HH:MM format (for example `23:00`).

Step d. To archive the data collected on SCB more than once a day, click ⊞ and enter the time for the other archive process. Repeat this step as needed.

Note
In case an archive process is not finished before the next one starts, the next archive process waits for the previous process to be completed.

Only closed audit-trail files are archived, for which **Retention time in days** has already elapsed.

Step e. SCB can access the remote server via different protocols. Select the one to use from the available protocols:

- *Only cleanup, no archiving*: Do not archive data to a server, simply delete the data that is older than **Retention time in days**.

Warning
No archiving permanently deletes all audit trails and data that is older than **Retention time in days** without creating a backup copy or an archive. Such data is irrecoverably lost. Use this option with care.

- *SMB/CIFS*: Server Message Block protocol used on Microsoft Windows Network.

Warning
The CIFS implementation of NetApp storage devices is not compatible with the CIFS implementation used in SCB, therefore it is not possible to create backups and archives from SCB to NetApp devices using the CIFS protocol (the operation fails with a similar error message: `/opt/scb/mnt/14719217504d41370514043/reports/2010": Permission denied (13) '2010/day/' rsync: failed to set times on`).

To overcome this problem, either:

- use the NFS protocol to access your NetApp devices, or
- use a backup device that has a CIFS implementation compatible with SCB, for example, Windows or Linux Samba.

- *NFS*: Network File System protocol.

Warning
When using the NFS protocol to create backups or archives, ensure that the files on the remote server are readable for the `www-data` user as well, because SCB uses this user to access remote backups and archives if needed.

Step f. Enter the IP address or the hostname of the remote server into the **Target settings > Target server** field (for example `backup.example.com`).

Step g. Fill the **Retention time in days** field. Only audit data older than this value is archived to the external server.

Note
The archived data is deleted from SCB.

Connection metadata is not deleted. Log files are not deleted at this point, but are rotated on a weekly basis.

Note
Starting with SCB 3.0, data about archived connections can be automatically deleted from the connection database. For details, see *Procedure 5.1.12, Configuring cleanup for the SCB connection database (p. 147)*.

Step h. SCB organizes the audit trails into directories based on the date or the protocol. The subdirectories are created directly into the archive directory. Select one of the following directory structures:

- **Protocol/Connection/Archive Date/**
- **Archive Date/Connection/Protocol/**
- **Connection Date/Protocol/Connection/**
- **Archive Date/**
- **Connection Date/**

For example, **Protocol/Connection/Archive Date** template will have create subdirectories for the audited protocols (that is, `ssh, rdp, telnet, vnc`), for the name of the connection policy, and finally, for the date (YEAR-MONTH-DAY in `YYYY-MM-DD` format).

Note
Connection Date refers to the time the connection started, while **Archive Date** to the time it was archived. The difference between the two dates depends on the retention time set for the archiving policy.

Step i. Provide the protocol-specific parameters for the selected method. The protocol-specific parameters are described in *Section 4.3.5.4, Parameters of the backup protocols (p. 70)*.

Step j. To receive e-mail notification of the backup, select the **Send notification on errors only** or the **Send notification on all events** option. Notifications are sent to the administrator e-mail address set on the **Management** tab, and include the list of the files that were backed up.

Note
This e-mail notification is different from the one set on the **Alerting & Monitoring** tab. This notification is sent to the administrator's e-mail address, while the alerts are sent to the alert e-mail address (see *Section 4.3.4, Configuring system monitoring on SCB (p. 57)*).

Step k. Click **Commit**.

Expected outcome:

An archive policy is created.

Step 2. To use this policy to archive the data of a connection policy, navigate to the connection (for example to **SSH Control > Connections**, select the connection to archive the recorded audit trails from, and select the archive policy you want to use in the **Archive/Cleanup policy** field.

Step 3. Click **Commit**.

Tip
To start the archiving process immediately, click **Archive now**. The **Archive now** functionality works only after the archiving has been configured.

4.3.5.3. Procedure – Encrypting configuration backups with GPG

Purpose:

The configuration file of SCB during system-backups, upload the public-part of a GPG key. To enable GPG-encryption, complete the following steps:

Warning
The regular backups of SCB contain other information (for example databases), but only the configuration file is encrypted. Note that system-backups do not contain data like audit-trails.

Steps:

Step 1. Navigate to **Basic > System > Management > System backup**.

Step 2. Select **Encrypt configuration**.

Step 3. Select .

- To upload a key file, click **Browse**, select the file containing the public GPG key, and click **Upload**. SCB accepts both binary and ASCII-armored GPG keys.
- To copy-paste the key from the clipboard, paste it into the **Key** field and click **Set**.

Note
The GPG key you upload must be permitted to encrypt data. Keys that can be used only for signing cannot be used to encrypt the configuration file.

Step 4. Click **Commit**.

Note
It is not possible to directly import a GPG-encrypted configuration into SCB, it has to be decrypted locally first.

For details on restoring configuration from a configuration backup, see *Procedure 11.8, Restoring configuration from a configuration backup (p. 245)*

4.3.5.4. Parameters of the backup protocols

This section describes the details of the protocols used for data backup and archiving. Parameters of the **NFS** protocol are described below.

- For details on using Rsync, see *Procedure 4.3.5.4.1, Configuring Rsync over SSH (p. 70)*.
- For details on using the Samba protocol, see *Procedure 4.3.5.4.2, Configuring SMB (p. 72)*.
- For details on using NFS, see *Procedure 4.3.5.4.3, Configuring NFS (p. 73)*.

4.3.5.4.1. Procedure – Configuring Rsync over SSH

Purpose:

The **Rsync over SSH** backup method connects the target server with SSH and executes the `rsync` UNIX command to copy the data to the remote server. SCB authenticates itself with a public key — password-based authentication is not supported. To configure this method, complete the following steps.

Warning
The backup server must run rsync version 3.0 or newer.

Steps:

Step 1. Select **Rsync over SSH** from the **Target settings** radio buttons.

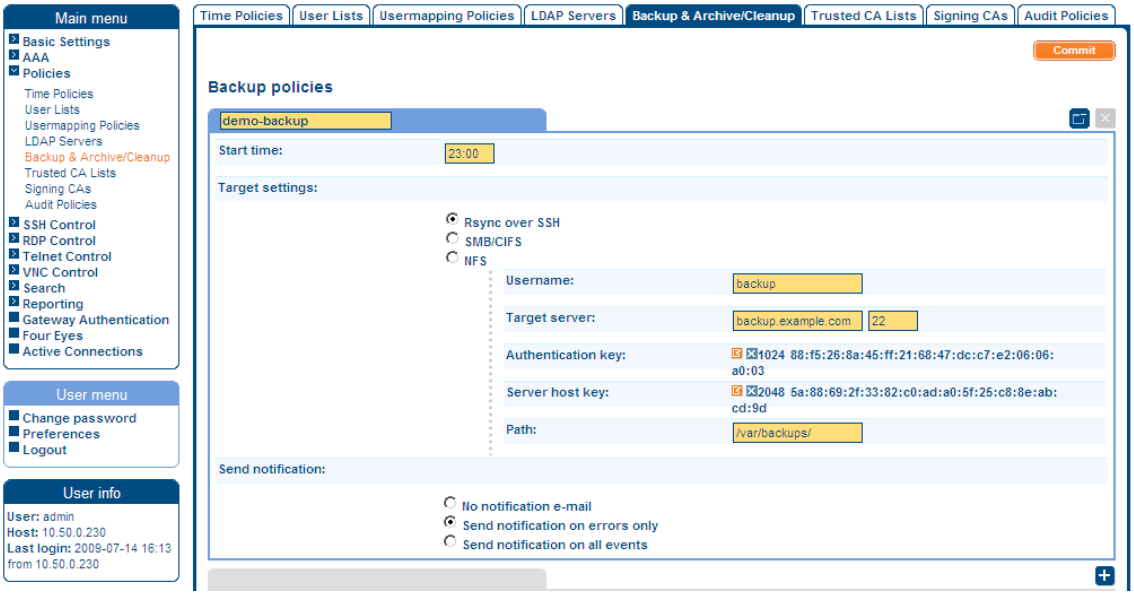


Figure 4.20. Configuring backups using rsync

Step 2. Enter the username used to logon to the remote server into the **Username** field.

Step 3. Click in the *Authentication key* field. A popup window is displayed.

Step 4. Generate a new keypair by clicking **Generate** or upload or paste an existing one. This key will be used to authenticate SCB on the remote server. The public key of this keypair must be imported to the remote server.

Step 5. Click in the *Server host key* field. A popup window is displayed.

Step 6.

Figure 4.21. Configuring SSH keys

Click **Query** to download the host key of the server, or upload or paste the host key manually. SCB will compare the host key shown by the server to this key, and connect only if the two keys are identical.

Step 7. Enter the port number of the SSH server running on the remote machine into the **Port** field.

Step 8. Enter the path to the backup directory on the target server into the **Path** field (for example `/backups`). SCB saves all data into this directory, automatically creating the subdirectories. Backups of audit-trails are stored in the `data`, configuration backups in the `config` subdirectory.

Step 9. Click **Commit**.

4.3.5.4.2. Procedure – Configuring SMB

Purpose:

The **SMB/CIFS** backup method connects to a share on the target server with Server Message Block protocol. SMB/CIFS is mainly used on Microsoft Windows Networks. To configure this method, complete the following steps.

Warning
The CIFS implementation of NetApp storage devices is not compatible with the CIFS implementation used in SCB, therefore it is not possible to create backups and archives from SCB to NetApp devices using the CIFS protocol (the operation fails with a similar error message: `/opt/scb/mnt/1471921750 4d41370514043/reports/2010": Permission denied (13) '2010/day/' rsync: failed to set times on`).

To overcome this problem, either:

- use the NFS protocol to access your NetApp devices, or
- use a backup device that has a CIFS implementation compatible with SCB, for example, Windows or Linux Samba.

Steps:

Step 1. Select **SMB/CIFS** from the **Target settings** radio buttons.

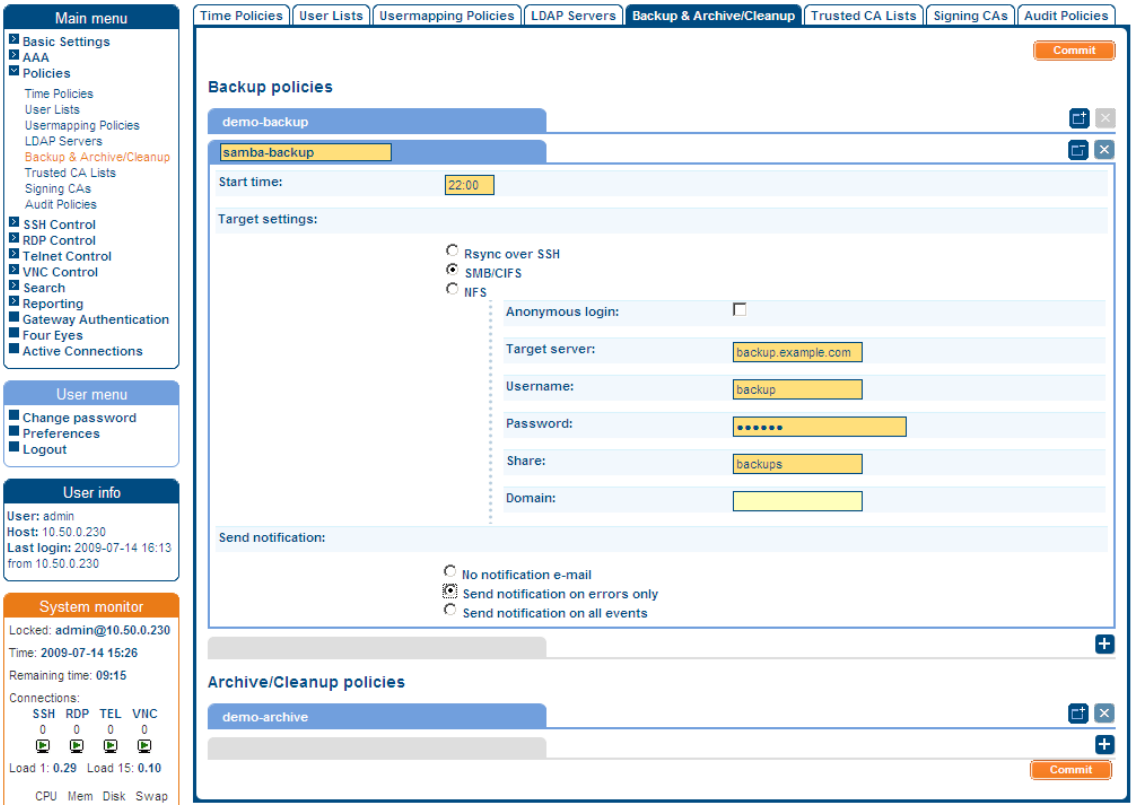


Figure 4.22. Configuring backups via SMB/CIFS

Step 2. Enter the username used to logon to the remote server into the **Username** field, or select the **Anonymous** option.

Step 3. Enter the password corresponding to the username into the **Password** field.

Step 4. Enter the name of the share into the **Share** field.
SCB saves all data into this directory, automatically creating the subdirectories. Backups of audit-trails are stored in the `data`, configuration backups in the `config` subdirectory.

Step 5. Enter the domain name of the target server into the **Domain** field.

Step 6. Click **Commit**.

4.3.5.4.3. Procedure – Configuring NFS

Purpose:

The **NFS** backup method connects to a shared directory of the target server with the Network File Share protocol. To configure this method, enter the name of the NFS export into the **Export** field. SCB saves all data into this directory, automatically creating the subdirectories. Audit-trail backups are stored in the `data`, configuration backups in the `config` subdirectory.

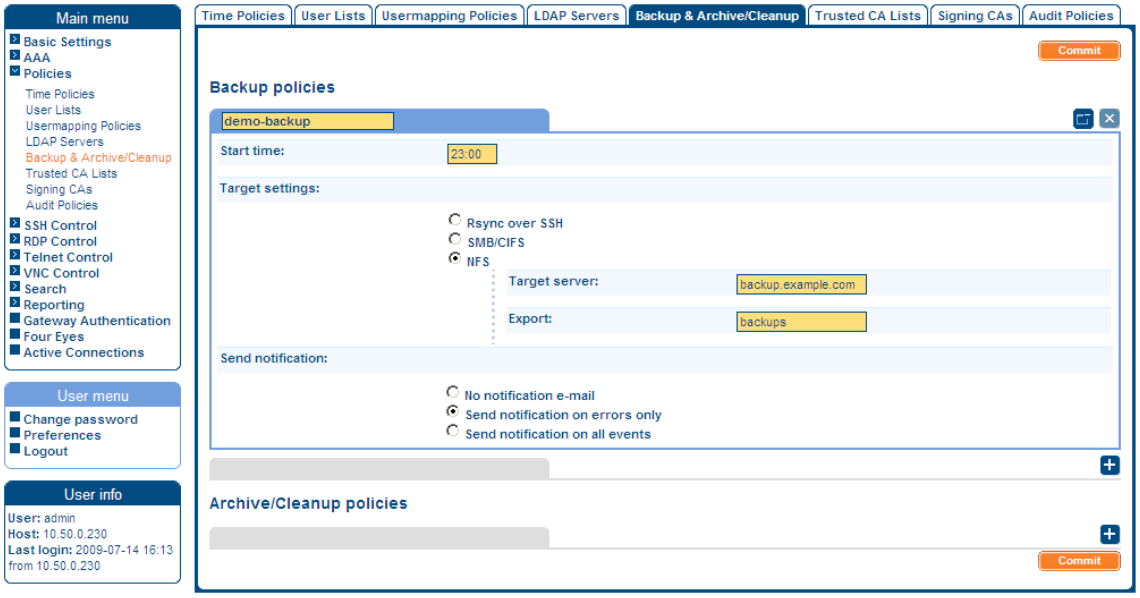


Figure 4.23. Configuring NFS backups

The backup server must also be configured to accept backups from SCB. To configure NFS on the remote server, complete the following steps:

Note
These steps must be performed on the remote server, not on SCB.

Steps:

Step 1. Add a line that corresponds to the settings of SCB to the `/etc/exports` file of the backup server. This line should contain the following parameters:

- The path to the backup directory as set in the **Export** field of the SCB backup or archiving policy.
- The IP address of the SCB interface that is used to access the remote server (that is, the address of the external interface, or the address of the management interface if it is enabled and the routing table of SCB is correctly configured — for details, see *Section 4.3.1, Network settings (p. 46)*.
- The following parameters: *(rw,no_root_squash,sync)*.

Example 4.1. Configuring NFS on the remote server
For example, if SCB connects the remote server from the *192.168.1.15* IP address and the data is saved into the */var/backups/SCB* directory, add the following line to the `/etc/exports` file:

```
/var/backups/SCB 192.168.1.15(rw,no_root_squash,sync)
```

Step 2. Execute the following command: `exportfs -a`

Step 3. Verify that the *`rpc portmapper`* and *`rpc.statd`* applications are running.

4.3.5.5. Ownership of the backup files

The different backup protocols assign different file ownerships to the files saved on the backup server. The owners of the backup files created using the different protocols are the following:

- *rsync*: The user provided on the web interface.
- *SMB*: `www-data`
- *NFS*: `root` with `no-root-squash`, `nobody` otherwise.

Warning
SCB cannot modify the ownership of a file that already exists on the remote server. If you change the backup protocol but you use the same directory of the remote server to store the backups, make sure to adjust the ownership of the existing files according to the new protocol. Otherwise SCB cannot overwrite the files and the backup procedure fails.

4.4. User management and access control

The **AAA** menu (Authentication, Authorization, and Accounting) allows you to control the authentication, authorization, and accounting settings of the users accessing SCB. The following will be discussed in the next sections:

- For details on how to authenticate locally on SCB — see *Procedure 4.4.1, Managing SCB users locally (p. 75)*.
- For details on how to authenticate users using an external LDAP (for example Microsoft Active Directory) database — see *Procedure 4.4.4, Managing SCB users from an LDAP database (p. 79)*.
- For details on how to authenticate users using an external RADIUS server — see *Procedure 4.4.5, Authenticating users to a RADIUS server (p. 82)*.
- For details on how to control the privileges of users and usergroups — see *Section 4.4.6, Managing user rights and usergroups (p. 84)*.
- For details on how to display the history of changes of SCB configuration — see *Section 4.4.7, Listing and searching configuration changes (p. 90)*.

4.4.1. Procedure – Managing SCB users locally

Purpose:

By default, SCB users are managed locally on SCB. To create and delete local users, modify the group membership of local users, or to modify the password of a user, complete the following procedure.

Note
The `admin` user is available by default and has all possible privileges. It is not possible to delete this user.

Local users cannot be managed when LDAP authentication is used (see *Procedure 4.4.4, Managing SCB users from an LDAP database (p. 79)*). When LDAP authentication is enabled, the accounts of local users is disabled, they are not displayed on the **AAA > Local Users** page, but they are not deleted,

When using RADIUS authentication together with local users, the users are authenticated to the RADIUS server, only their group memberships must be managed locally on SCB. For details, see *Procedure 4.4.5, Authenticating users to a RADIUS server (p. 82)*.

Steps:

Step 1. Navigate to **AAA > Local Users** and click ⊞.

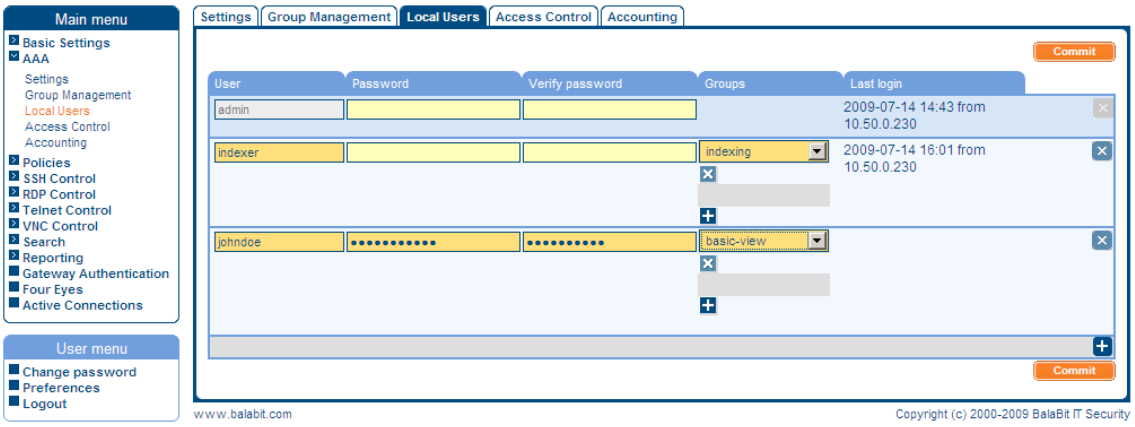


Figure 4.24. Creating local users

Step 2. Enter the username into the **User** field.

Note
The following characters cannot be used in usernames: `\/[]:;|=,+*?<>`

Step 3. Enter a password for the user into the **Password** and **Verify password** fields.
The strength of the password is indicated below the **Password** field as you type. To set a policy for password strength, see *Procedure 4.4.2, Setting password policies for local users (p. 76)*. The user can change the password later from the SCB web interface.

Step 4. Click ⊞ in the **Groups** section and select a group that the user will be member of. Repeat this step to add the user to multiple groups. For details about the different groups, see *Section 4.4.6, Managing user rights and usergroups (p. 84)*.

- To remove a user from a group, click ⊠ next to the group.
- To delete a user, click ⊠ at the right edge of the screen.

Step 5. Click **Commit**.

4.4.2. Procedure – Setting password policies for local users

Purpose:

SCB can use password policies to enforce minimal password strength and password expiry. To create a password policy, complete the following steps.

Warning
When running the Audit Player application in service mode, the Audit Player application needs a valid user account to access SCB. When using password expiry, ensure that the password of this user is changed in time, and that it is changed also on the Audit Player hosts.

Note
Password policies apply only for locally managed users, it has no effect if you manage your users from an LDAP database, or if you authenticate your users to a RADIUS server.

Password policies do not apply to the built-in `admin` user.

Steps:

Step 1. Navigate to **AAA > Settings**.

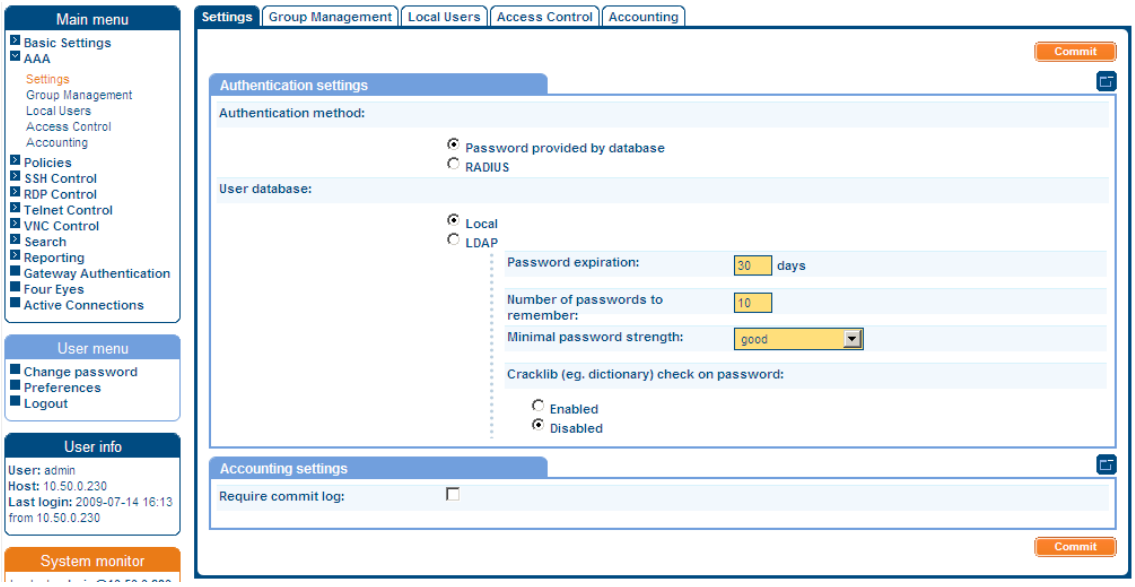


Figure 4.25. Configuring password policies

Step 2. Verify that the **Authentication method** is set to **Password provided by database** and that the **User database** is set to **Local**.

Note
If the setting of these fields is different (for example LDAP or RADIUS), then SCB is not configured to manage passwords locally.

Step 3. Set how long the passwords are valid in the **Password expiration** field. After this period, SCB users will have to change their password. To disable password expiry, enter `0`.

Step 4. To prevent password-reuse (for example when a user has two password and instead of changing to a new password only switches between the two), set how many different passwords must the user use before reusing an old password.

Step 5. To enforce the use of strong password, select the level of password-complexity from the **Minimal password strength** field.

Note
The strength of the password is determined by its entropy: the variety of numbers, letters, capital letters, and special characters used, not only by its length.

The **Enable cracklib** option executes some simple dictionary-based attacks to find weak passwords.

Step 6. Click **Commit**.

Note
Changes to the password policy do not affect existing passwords. However, setting password expiry will require every user to change their passwords after the expiry date, and the new passwords must comply with the strength requirements set in the password policy.

4.4.3. Procedure – Managing local usergroups

Purpose:

To display which users belong to a particular local usergroup, navigate to **AAA > Group Management**. You can edit the group memberships here as well.

You can use local groups to control the privileges of SCB local and LDAP users — who can view and configure what.

For the description of built-in groups, see *Section 4.4.6.5, Built-in usergroups of SCB (p. 88)*. To create a new group, complete the following steps:

Steps:

Step 1. Navigate to **AAA > Group Management** and click ⊞.

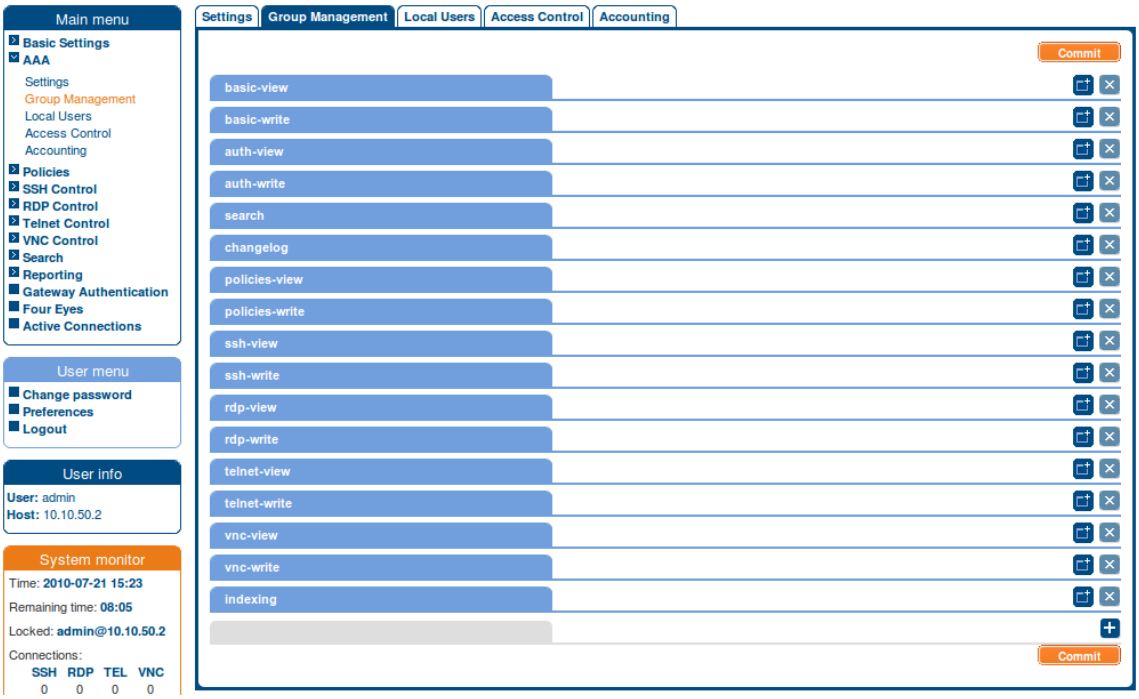


Figure 4.26. Group management

Step 2. Enter a name for the group.

Step 3. Enter the names of the users belonging to the group. Click ⊞ to add more users.

Step 4. Click **Commit**.

4.4.4. Procedure – Managing SCB users from an LDAP database

Purpose:

The SCB web interface can authenticate users to an external LDAP database to simplify the integration of SCB to your existing infrastructure. You can also specify multiple LDAP servers; if the first server is unavailable, SCB will try to connect to the second server. To enable LDAP authentication, complete the following steps.

Note

The `admin` user is available by default and has all privileges. It is not possible to delete this user.

The `admin` user can login to SCB even if LDAP authentication is used.

Enabling LDAP authentication automatically disables the access of every local user except for `admin`.

SCB accepts both pre-win2000-style and Win2003-style account names (User Principal Names). User Principal Names (UPNs) consist of a username, the at (@) character, and a domain name, for example `administrator@example.com`.

The following characters cannot be used in usernames and group names: `/\[]:;|=,+*)?<>@"`

When using RADIUS authentication together with LDAP users, the users are authenticated to the RADIUS server, only their group memberships must be managed in LDAP. For details, see *Procedure 4.4.5, Authenticating users to a RADIUS server (p. 82)*.

Steps:

Step 1. Navigate to **AAA > Settings > Authentication settings**.

Step 2. Select the **LDAP** option and enter the parameters of your LDAP server.

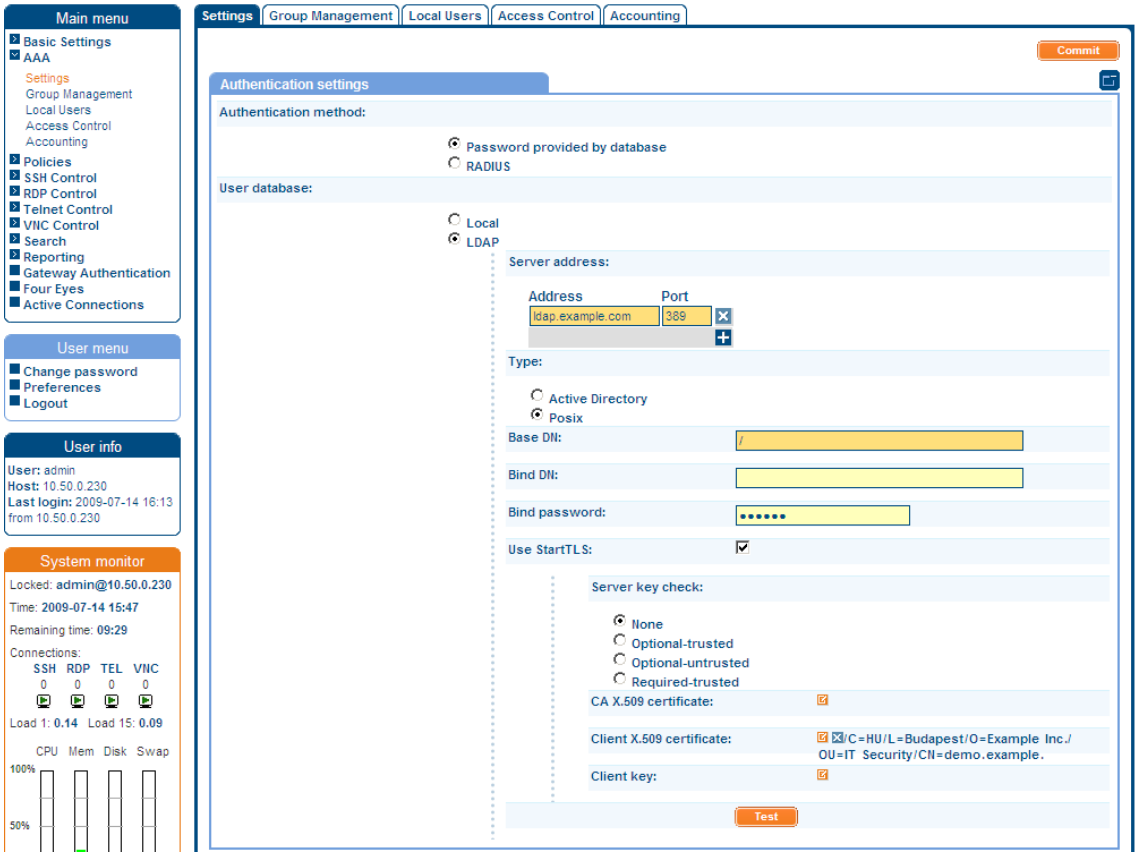


Figure 4.27. Configuring LDAP authentication

Step a. Enter the IP address or hostname and port of the LDAP server into the **Server Address** field.

To add multiple servers, click ⊞ and enter the address of the next server. If a server is unreachable, SCB will try to connect to the next server in the list in failover fashion.

Warning
If you will use a TLS-encrypted with certificate verification to connect to the LDAP server, use the full domain name (for example `ldap.example.com`) in the **Server Address** field, otherwise the certificate verification might fail. The name of the LDAP server must appear in the Common Name of the certificate.

Step b. Select the type of your LDAP server in the **Type** field. Select **Active Directory** to connect to Microsoft Active Directory servers, or **Posix** to connect to servers that use the POSIX LDAP scheme.

Step c. Enter the name of the DN to be used as the base of the queries into the **Base DN** field (for example `DC=demodomain,DC=exampleinc`).

Step d. Enter the name of the DN where SCB should bind to before accessing the database into the **Bind DN** field.
For example: `CN=Administrator,CN=Users,DC=demodomain,DC=exampleinc`.

Note
SCB accepts both pre-win2000-style and Win2003-style account names (User Principal Names), for example `administrator@example.com` is also accepted.

Step e. Enter the password to use when binding to the LDAP server into the **Bind Password** field.

Step 3. If you want to encrypt the communication between SCB and the LDAP server, select the **Use StartTLS** option and complete the following steps:

Note
TLS-encrypted connection to Microsoft Active Directory is supported only on Windows 2003 Server and newer platforms. Windows 2000 Server is not supported.

Only the STARTTLS method is supported to encrypt the communication between the LDAP server and SCB. The LDAPS protocol is not supported.

- If you want SCB to verify the certificate of the server, click the icon in the **CA X.509 certificate** field. A popup window is displayed.

  Click **Browse**, select the certificate of the Certificate Authority (CA) that issued the certificate of the LDAP server, then click **Upload**. Alternatively, you can paste the certificate into the **Copy-paste** field and click **Set**.

  SCB will use this CA certificate to verify the certificate of the server, and reject the connections if the verification fails.

  If you want SCB to simply accept any certificate shown by the LDAP server, select **Required untrusted** in the **Server key check** field.

Note
Alternatively, you can use the following options to check the certificate of the LDAP server:

- *None*: Do not request a certificate from the remote host, and accept any certificate if the host sends one.
- *Optional trusted*: If the remote host sends a certificate, SCB checks if it is valid (not expired) and that the Common Name of the certificate contains the domain name or the IP address of the host. If these checks fail, SCB rejects the connection. However, SCB accepts the connection if the host does not send a certificate.
- *Optional untrusted*:Accept any certificate shown by the remote host. Note that the host must show a certificate.
- *Required trusted*: Verify the certificate of the remote host. Only valid certificates signed by a trusted certificate authority are accepted. For details on importing CA certificates, see *Procedure 4.5.7.2, Uploading external certificates to SCB (p. 118)*. Note that the Common Name of the certificate must contain the domain name or the IP address of the host.

- *Request untrusted*: SCB requests a certificate from the remote host, and rejects the connection if no certificate is received; if the certificate is not valid (expired); or if the Common Name of the certificate does not contain the domain name or the IP address of the host.

Warning
If you will use a TLS-encrypted with certificate verification to connect to the LDAP server, use the full domain name (for example `ldap.example.com`) in the **Server Address** field, otherwise the certificate verification might fail. The name of the LDAP server must appear in the Common Name of the certificate.

- If the LDAP server requires mutual authentication, that is, it expects a certificate from SCB, generate and sign a certificate for SCB, then click in the **Client X.509 certificate** field to upload the certificate. After that, click in the **Client key** field and upload the private key corresponding to the certificate.

Step 4. If you have modified the **Server key check** field or the keys used in the connections, perform the following steps:

Warning
This step terminates all controlled connections going through SCB. Disconnect your clients from the protected servers before proceeding.

To activate the new settings, navigate to **Basic Settings > System > System control**, and click **All services > Restart**.

Step 5. Click **Commit**.

Note
You also have to configure the usergroups in SCB and possibly in your LDAP database. For details on using usergroups, see *Section 4.4.6.4, How to use usergroups (p. 87).*

Step 6. Click **Test** to test the connection. Note that the testing of SSL-encrypted connections is currently not supported.

4.4.5. Procedure – Authenticating users to a RADIUS server

Purpose:

SCB can authenticate its users to an external RADIUS server. Group memberships of the users must be managed either locally on SCB or in an LDAP database.

Warning
The challange/response authentication methods is currently not supported. Other authentication methods (for example password, SecureID) should work.

To authenticate SCB users to a RADIUS server, complete the following steps:

Steps:

Step 1. Navigate to **AAA > Settings**.

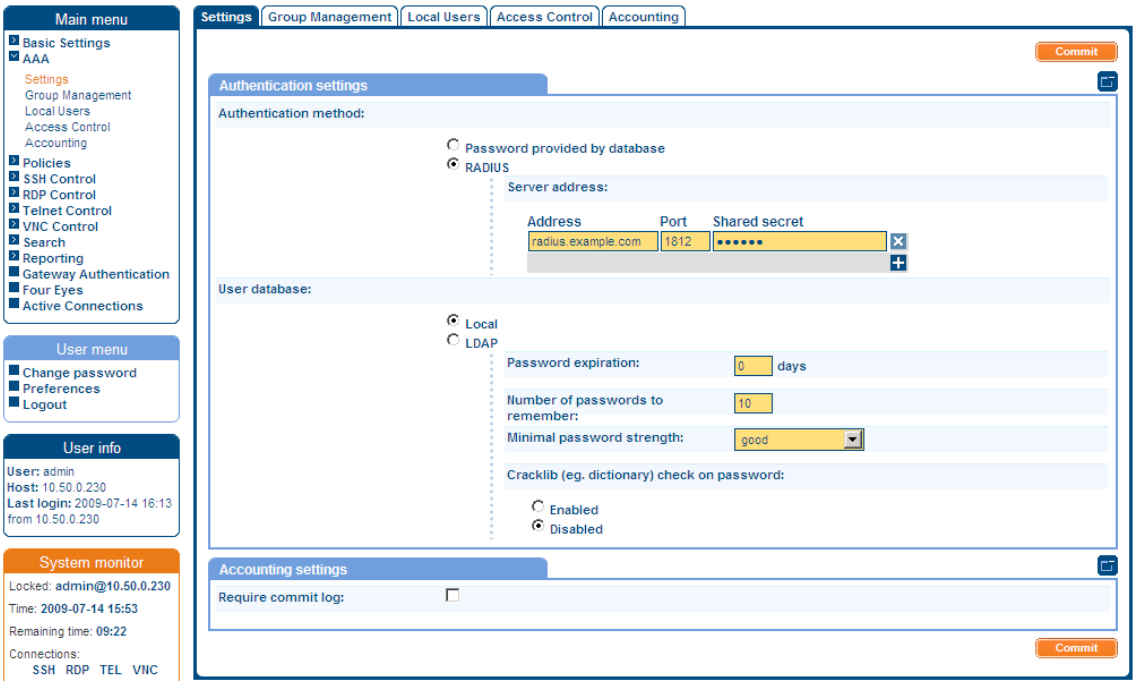


Figure 4.28. Configuring RADIUS authentication

Step 2. Set the **Authentication method** field to **RADIUS**.

Step 3. Enter the IP address or domain name of the RADIUS server into the **Address** field.

Step 4. Enter the password that SCB can use to access the server into the **Shared secret** field.

Step 5. To add more RADIUS servers, click ✚ and repeat Steps 2-4.
Repeat this step to add multiple servers. If a server is unreachable, SCB will try to connect to the next server in the list in failover fashion.

Step 6. When configuring RADIUS authentication with a local user database, complete the following steps.

Step a. Set **Password expiration** to `0`.

Step b. Set **Number of passwords to remember** to `0`.

Step c. Set **Minimal password strength** to `disabled`.

Step d. Set **Cracklib check on password** to `disabled`.

Step 7.

Warning
After clicking **Commit**, the SCB web interface will be available only after successfully authenticating to the RADIUS server. Note that the default `admin` account of SCB will be able to login normally, even if the RADIUS server is unaccessible.

Click **Commit**.

4.4.6. Managing user rights and usergroups

In SCB, user rights can be assigned to usergroups. SCB has numerous usergroups defined by default, but custom user groups can be defined as well. Every group has a set of privileges: which pages of the SCB web interface it can access, and whether it can only view (read) or also modify (read & write/perform) those pages or perform certain actions.

Note
Every group has either read or read & write/perform privileges to a set of pages.

- For details on modifying existing groups, see *Procedure 4.4.6.1, Modifying group privileges (p. 85)*.
- For details on creating a new usergroup, see *Procedure 4.4.6.2, Creating new usergroups for the SCB web interface (p. 86)*.
- For details on finding usergroups that have a specific privilege, see *Section 4.4.6.3, Finding specific usergroups (p. 87)*.
- For tips on using usergroups, see *Section 4.4.6.4, How to use usergroups (p. 87)*.
- For a detailed description about the privileges of the built-in usergroups, see *Section 4.4.6.5, Built-in usergroups of SCB (p. 88)*.

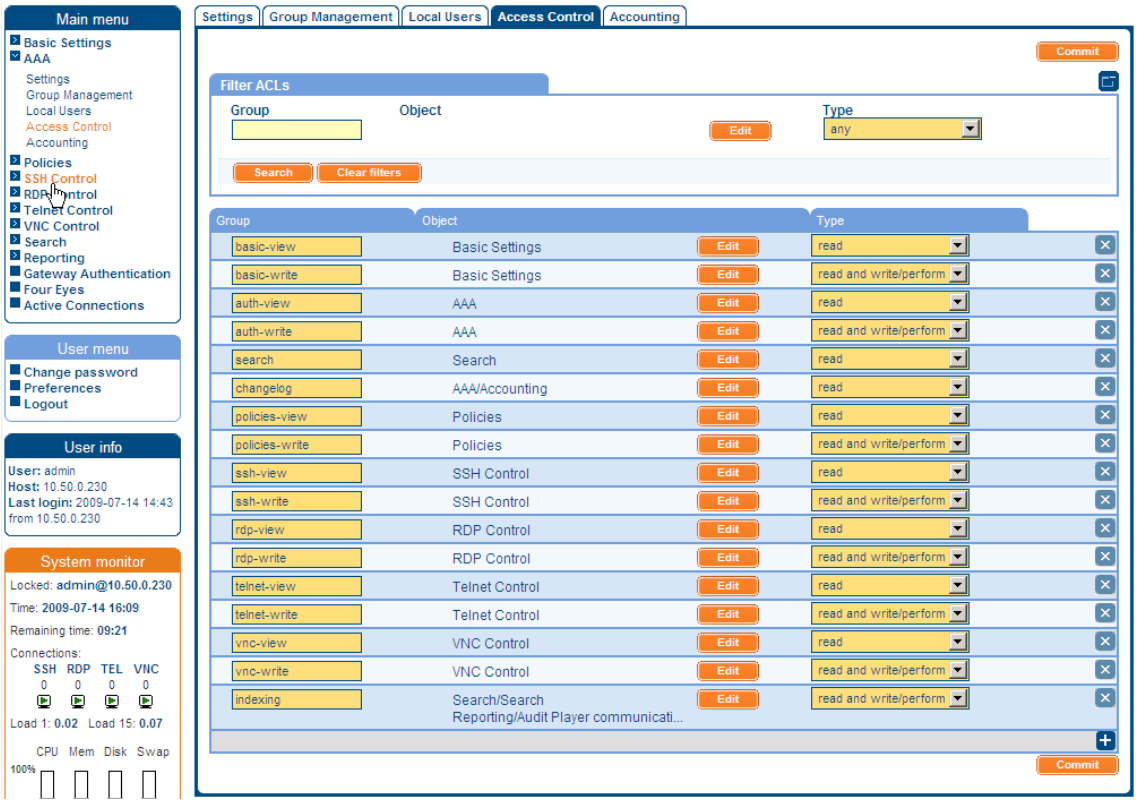


Figure 4.29. Managing SCB users

4.4.6.1. Procedure – Modifying group privileges

Purpose:

To modify the privileges of an existing group, complete the following steps:

Steps:

Step 1. Navigate to **AAA > Access Control**.

Step 2. Find the group you want to modify and click **Edit**. The list of available privileges is displayed.

Step 3. Select the privileges (pages of the SCB interface) to which the group will have access to and click **Save**.

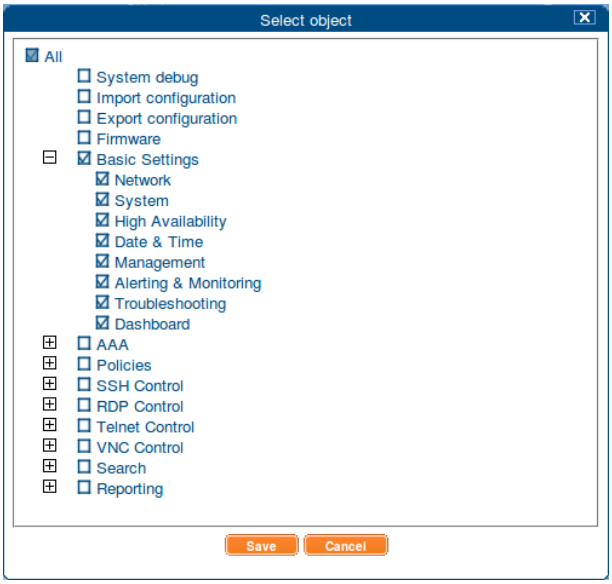


Figure 4.30. Modifying group privileges

Warning
Assigning the **Search** privilege to a user on the AAA page automatically enables the **Search in all connections** privilege, and grants the user access to every audit trail, even if the user is not a member of the groups listed in the **Access control** option of the particular connection policy.

Step 4. Select the type of access (read or read & write) from the **Type** field.

Step 5. Click **Commit**.

4.4.6.2. Procedure – Creating new usergroups for the SCB web interface

Purpose:

To create a new group, complete the following steps:

Steps:

Step 1. Navigate to **AAA > Access Control** and click .

Step 2. Enter a name for the group. For details on how you should name your groups, see *Section 4.4.6.4, How to use usergroups (p. 87)*.

Step 3. Click the icon located next to the name of the group. The list of available privileges is displayed.

Step 4. Select the privileges (pages of the SCB interface) to which the group will have access to and click **Save**.

Step 5. Select the type of access (read or read & write) from the **Type** field.

Step 6. Click **Commit**.

Note
To export the configuration of SCB, the Export configuration privilege is required.

To import a configuration to SCB, the Import configuration privilege is required.

To update the firmware and set the active firmware, the Firmware privilege is required.

The *admin* user is available by default and has all privileges. It is not possible to delete this user.

4.4.6.3. Finding specific usergroups

The **Filter ACLs** section of the **AAA > Access Control** page provides you with a simple searching and filtering interface to search the names and privileges of usergroups.

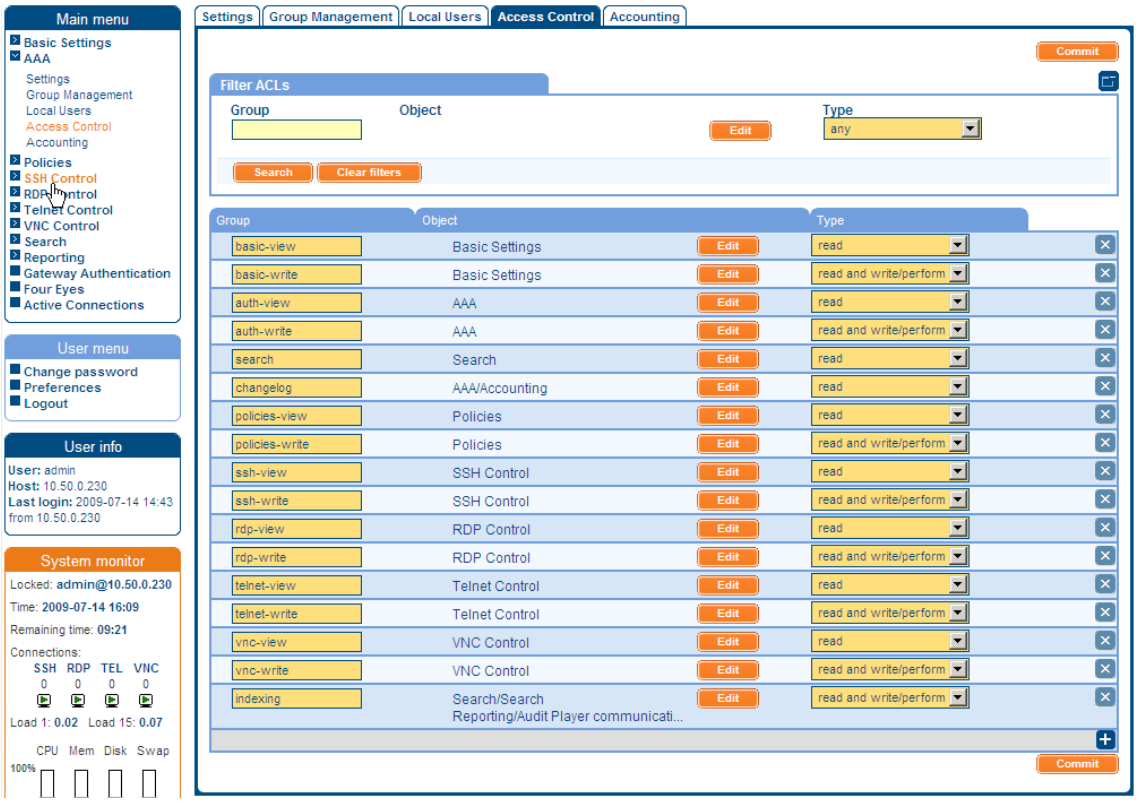


Figure 4.31. Finding specific usergroups

- To select usergroups starting with a specific string, enter the beginning of the name of the group into the **Group** field and select **Search**.
- To select usergroups who have a specific privilege, click , select the privilege or privileges you are looking for, and click **Search**.
- To filter for read or write access, use the **Type** option.

4.4.6.4. How to use usergroups

How you should name usergroups depends on the way you manage your SCB users.

- *Local users*: If you use only local users, create or modify your usergroups on the **AAA > Access Control** page and add users to the groups on the **AAA > Local Users** or the **AAA > Group Management** page.
- *LDAP users and LDAP groups*: If you manage your users from LDAP, and also have LDAP groups that match the way you want to group your SCB users, create or modify your usergroups on the **AAA > Access Control** page and ensure that the name of your LDAP group and the SCB usergroup is the same. For example, to make members of the `admins` LDAP group be able to use SCB, create a usergroup called `admins` on the **AAA > Access Control** page and edit the privileges of the group as needed.
- *RADIUS users and local groups*: This is the case when you manage users from RADIUS, but you cannot or do not want to create groups in LDAP. Create your local groups on the **AAA > Access Control** page, and add your RADIUS users to these groups on the **AAA > Group Management** page.

4.4.6.5. Built-in usergroups of SCB

SCB has the following usergroups by default:

Figure 4.32. Built-in usergroups of SCB

Warning
If you use LDAP authentication on the SCB web interface and want to use the default usergroups, you have to create these groups in your LDAP database and assign users to them. For details on using usergroups, see *Section 4.4.6.4, How to use usergroups (p. 87)*.

- *basic-view*: View the settings in the **Basic Settings** menu, including the system logs of SCB. Members of this group can also execute commands on the **Troubleshooting** tab.
- *basic-write*: Edit the settings in the **Basic Settings** menu. Members of this group can manage SCB as a host.
- *auth-view*: View the names and privileges of the SCB administrators, the configured usergroups, and the authentication settings in the **AAA** menu. Members of this group can also view the history of configuration changes.
- *auth-write*: Edit authentication settings and manage users and usergroups.

Warning
Members of the `auth-write` group, or any other group with write privileges to the **AAA** menu are essentially equivalent to system administrators of SCB, because they can give themselves any privilege. Users with limited rights should never have such privileges.

If a user with write privileges to the **AAA** menu gives himself new privileges (for example gives himself group membership to a new group), then he has to relogin to the SCB web interface to activate the new privilege.

- *search*: Browse and download various logs and alerts in the **Search** menu. The members of this group have access to the audit trail files as well. Note that to open encrypted audit trail files, the proper encryption keys are required.

Note
In SCB version 1.x, a separate privilege was required to view and download audit trail files. Now members of the search group automatically have these privileges.

- *changelog*: View the history of SCB configuration changes in the **AAA > Accounting** menu.
- *report*: Browse, create and manage reports, and add statistics-based chapters to the reports in the **Reports** menu.

Note
To control exactly which statistics-based chapters and reports can the user include in a report, use the `Use static subchapters` privileges.

- *policies-view*: View the policies and settings in the **Policies** menu.
- *policies-write*: Edit the policies and settings in the **Policies** menu.
- *ssh-view*: View all connection and policy settings in the **SSH Control** menu.
- *ssh-write*: Edit all connection and policy settings in the **SSH Control** menu.
- *rdp-view*: View all connection and policy settings in the **RDP Control** menu.
- *rdp-write*: Edit all connection and policy settings in the **RDP Control** menu.
- *telnet-view*: View all connection and policy settings in the **Telnet Control** menu.

- *telnet-write*: Edit all connection and policy settings in the **Telnet Control** menu.
- *vnc-view*: View all connection and policy settings in the **VNC Control** menu.
- *vnc-write*: Edit all connection and policy settings in the **VNC Control** menu.
- *indexing*: Allows host running the Audit Player application to access and download audit trails for automatic indexing. Note that the members of this group can access the SCB web interface as well, and download any audit trail directly.

4.4.7. Listing and searching configuration changes

SCB automatically tracks every change of its configuration. To display the history of changes, select **AAA > Accounting**. The changes are organized as log messages, and can be browsed and searched using the regular SCB search interface (for details, see *Chapter 6, Browsing log messages and SCB reports (p. 175)*). The following information is displayed about each modification:

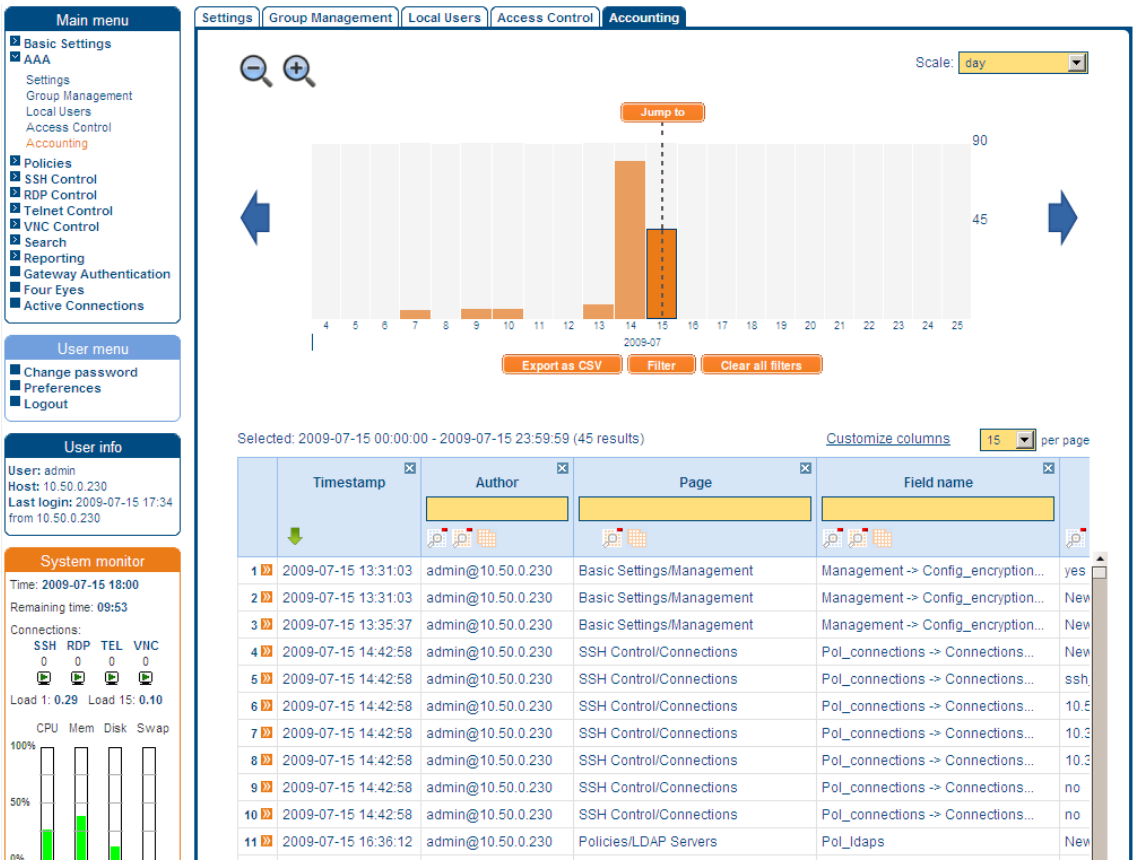


Figure 4.33. Browsing configuration changes

- *Timestamp*: The date of the modification.
- *Author*: Username of the administrator who modified the configuration of SCB.
- *Page*: The menu item that was modified.
- *Field name*: The name of the field or option that was modified.
- *New value*: The new value of the configuration parameter.

- *Message*: The changelog or commit log that the administrator submitted. This field is available only if the **Require commit log** option is enabled (see below).
- *Old value*: The old value of the configuration parameter.

To request the administrators to write an explanation to every configuration change, navigate to **AAA > Settings > Accounting settings** and select the **Require commit log** option.

4.5. Managing SCB

The following sections explain the basic management tasks of SCB, including the basic control (for example, shutdown or reboot) of the appliance, upgrading, as well as tips on troubleshooting SCB.

4.5.1. Controlling SCB — restart, shutdown

To restart or shut down SCB, navigate to **Basic Settings > System > System control > This node** and click the respective action button. The **Other node** refers to the slave node of a high availability SCB cluster. For details on high availability clusters, see *Section 4.5.2, Managing a high availability SCB cluster (p. 93)*.

Warning

- When rebooting the nodes of a cluster, reboot the other (slave) node first to avoid unnecessary takeovers.
- When shutting down the nodes of a cluster, shut down the other (slave) node first. When powering on the nodes, start the master node first to avoid unnecessary takeovers.
- When both nodes are running, avoid interrupting the connection between the nodes: do not unplug the Ethernet cables, reboot the switch or router between the nodes (if any), or disable the HA interface of SCB.

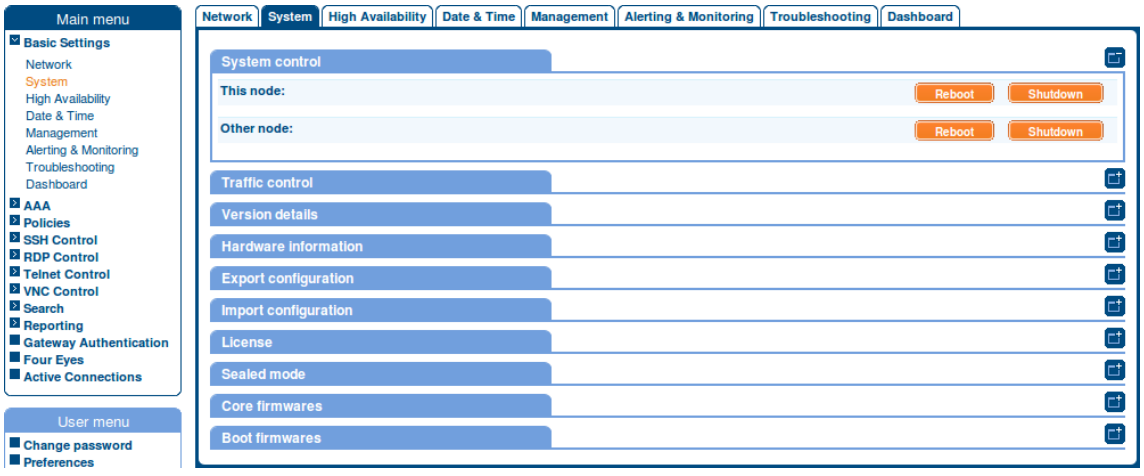


Figure 4.34. Performing basic management

Note

Web sessions to the SCB interface are persistent and remain open after rebooting SCB, so you do not have to relogin after a reboot.

4.5.1.1. Procedure – Disabling controlled traffic

Purpose:

To temporarily disable some or all of the controlled traffic to the protected servers, complete the following steps:

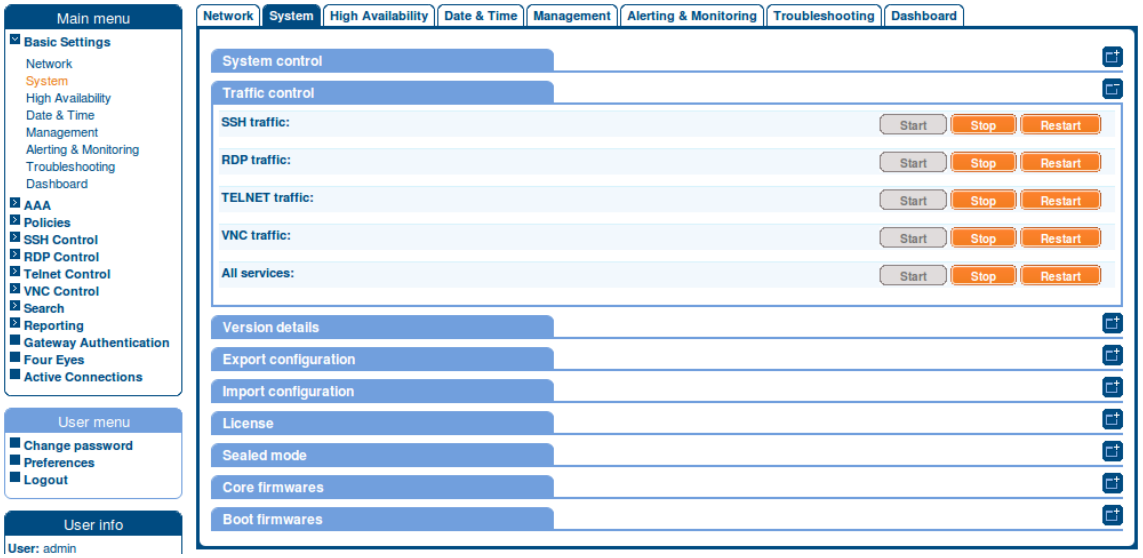


Figure 4.35. Disabling the controlled traffic

Warning
Disabling traffic that way is only temporary; connections will be enabled again after committing any other change from the SCB web interface. For details on how to permanently disable a type of traffic, see *Procedure 4.5.1.2, Disabling controlled traffic permanently (p. 93).*

Note
Disabling the traffic affects only the traffic configured in the Connection policies, other traffic can pass SCB even if the all traffic is disabled. For details on configuring Connection policies, see *Chapter 5, Configuring connections (p. 120).*

Steps:

Step 1. Navigate to the **Basic Settings > System > Traffic control**.

Step 2.
- To disable SSH traffic, click **Stop** in the **SSH traffic** field. Note that this also stops all other traffic forwarded in SSH, for example X11.
- To disable RDP traffic, click **Stop** in the **RDP traffic** field.
- To disable Telnet and TN3270 traffic, click **Stop** in the **Telnet traffic** field.
- To disable VNC traffic, click **Stop** in the **VNC traffic** field.
- To disable all types of traffic, click **Stop** in the **All traffic** field.

The **System monitor** displays the status of all types of traffic.

4.5.1.2. Procedure – Disabling controlled traffic permanently

Note
Disabling the traffic affects only the traffic configured in the Connection policies, other traffic can pass SCB even if the all traffic is disabled. For details on configuring Connection policies, see *Chapter 5, Configuring connections (p. 120)*.

Steps:

Step 1.

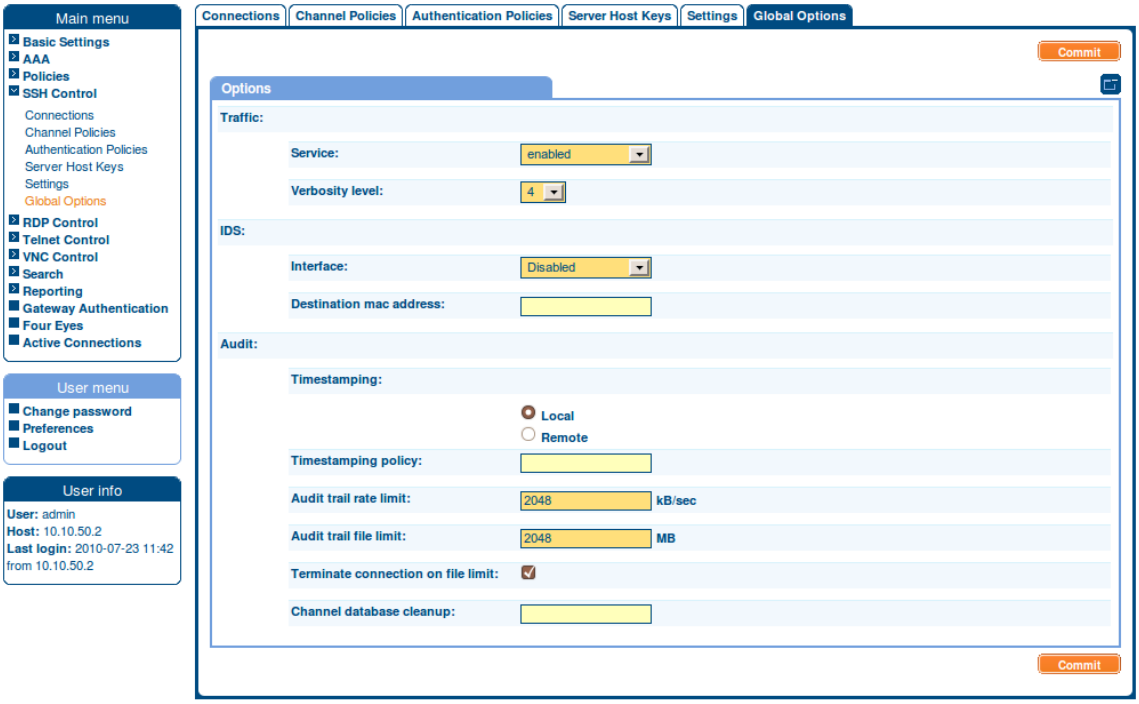


Figure 4.36. Disabling the controlled traffic persistently

Navigate to the **Global Options** page of the traffic type you want to disable, for example to **SSH Control > Global Options** to disable SSH traffic.

Step 2. Set the **Traffic** field to `disabled`.

Step 3. Click **Commit**.

4.5.2. Managing a high availability SCB cluster

Warning

- When rebooting the nodes of a cluster, reboot the other (slave) node first to avoid unnecessary takeovers.
- When shutting down the nodes of a cluster, shut down the other (slave) node first. When powering on the nodes, start the master node first to avoid unnecessary takeovers.
- When both nodes are running, avoid interrupting the connection between the nodes: do not unplug the Ethernet cables, reboot the switch or router between the nodes (if any), or disable the HA interface of SCB.

The **Basic Settings > High Availability** page provides the following information about SCB:

Note
Refer to *Procedure 3.2, Installing two SCB units in HA mode (p. 251)* of *Appendix 3, BalaBit Shell Control Box Hardware Installation Guide (p. 250)* for details on creating a high availability SCB cluster.

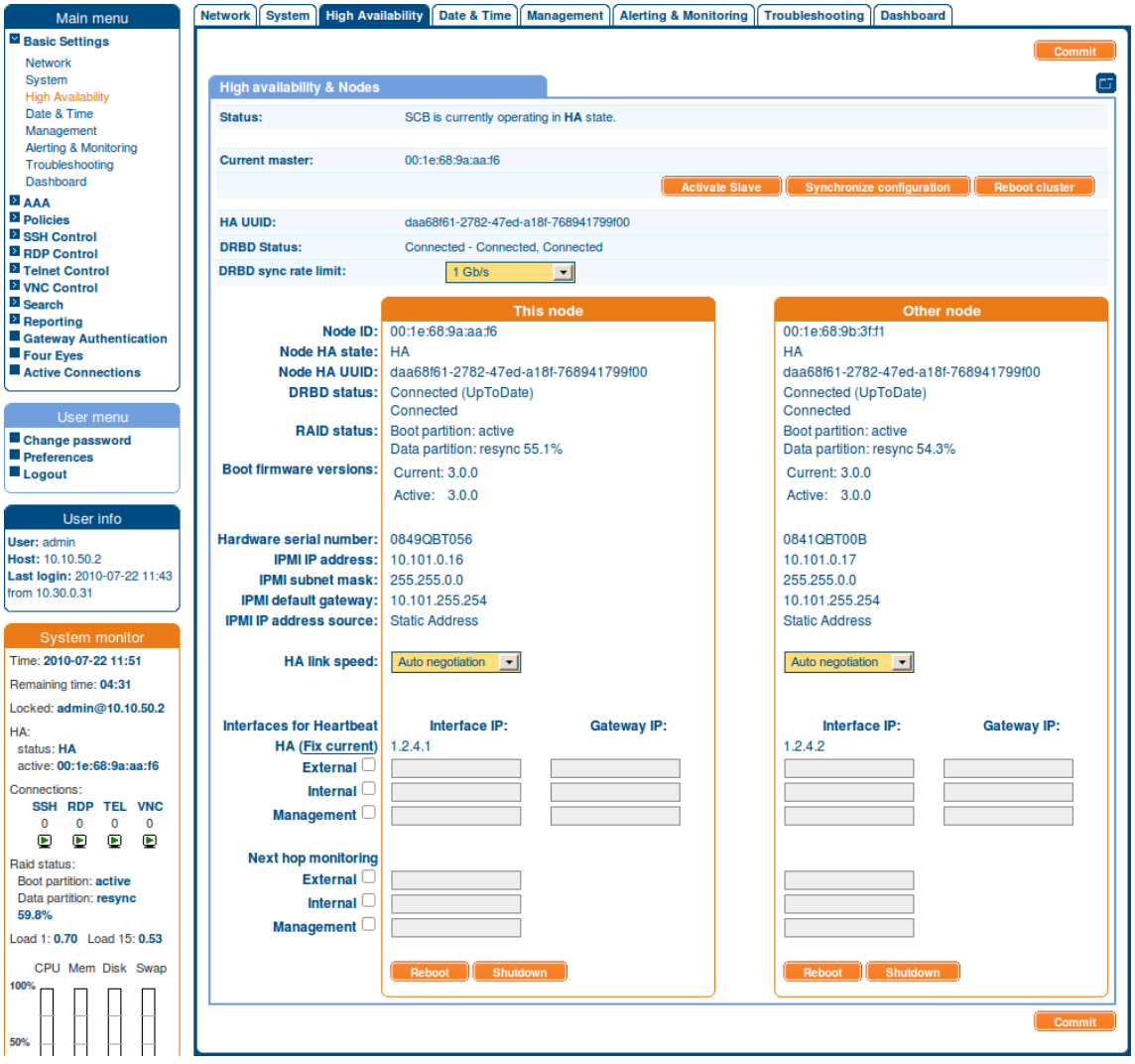


Figure 4.37. Managing a high availability cluster

- *Status*: Indicates whether SCB is running in High Availability or Standalone mode.
- *Node ID*: The MAC address of the `HA` interface of the node. This address is also printed on a label on the top cover of the SCB unit.
- *Node HA state*: Indicates whether the SCB node is running in High Availability or Standalone mode.
- *Node HA UUID*: A unique identifier of the node. Only available in High Availability mode.

- *DRBD status*: The status of data stored on SCB. The status must be `Consistent` on the active node to prevent data loss.
- *RAID status*: The status of the RAID device of the node.

The active (master) SCB node is labeled as **This node**, this unit inspects the SSH traffic and provides the web interface. The SCB unit labeled as **Other node** is the slave node that is activated if the master node becomes unavailable.

Note
For SCB clusters, the ID of the node (the MAC address of its HA interface) sending the message is included in the log messages. Note that if the central log server is a syslog-ng server, the `keep-hostname` option should be enabled on the syslog-ng server.

To activate the other node and disable the currently active node, click **Activate slave**.

To reboot both nodes, click **Reboot Cluster**. Note that this will automatically reboot the two nodes, but will not result in a takeover.

Warning
Hazard of data loss! Activating the slave node terminates all connections of SCB and might result in data loss. The slave node becomes active after about 60 seconds, during which the protected servers cannot be accessed.

4.5.2.1. High Availability status explained

This section explains the possible statuses of the SCB nodes and the DRBD data storage system. SCB displays this information on the **Basic Settings > High Availability** page.

Note
If a redundant Heartbeat interface is configured, its status is also displayed in the **Redundant Heartbeat status** field. For details on redundant Heartbeat interfaces, see *Section 4.5.2.4, Redundant Heartbeat interface status explained (p. 101)* and *Procedure 4.5.2.3, Configuring redundant Heartbeat interfaces (p. 98)*.

The **Status** field indicates whether the SCB nodes recognize each other properly and whether those are configured to operate in high availability mode. The status of the individual SCB nodes is indicated in the **Node HA status** field of the each node. The following statuses can occur:

- *Standalone*: There is only one SCB unit running in `standalone` mode, or the units have not been converted to a cluster (the **Node HA status** of both nodes is `standalone`). Click **Convert to Cluster** to enable High Availability mode.
- *HA*: The two SCB nodes are running in High Availability mode. **Node HA status** is `HA` on both nodes, and the **Node HA UUID** is the same on both nodes.
- *Half*: High Availability mode is not configured properly, one node is in `standalone`, the other one in `HA` mode. Connect to the node in `HA` mode, and click **Join HA** to enable High Availability mode.

- *Broken*: The two SCB nodes are running in High Availability mode. **Node HA status** is `HA` on both nodes, but the **Node HA UUID** is different. Contact the BalaBit Support Team for help. For contact details, see *Section 5, Contact and support information (p. xiv)*.
- *Degraded*: SCB was running in high availability mode, but one of the nodes has disappeared (for example broken down, or removed from the network). Power on, reconnect, or repair the missing node.
- *Degraded Sync*: Two SCB units were joined to High Availability mode, and the first-time synchronization of the disks is currently in progress. Wait for the synchronization to complete. Note that in case of large disks with lots of stored data, synchronizing the disks can take several hours.
- *Split brain*: The two nodes lost the connection to each other, with the possibility of both nodes being active (master) for a time.

Warning
Hazard of data loss! In this case, valuable audit trails might be available on both SCB nodes, so special care must be taken to avoid data loss. For details on solving this problem, see *Procedure 11.7.2, Recovering from a split brain situation (p. 243)*.

Do NOT reboot or shut down the nodes.

- *Invalidated*: The data on one of the nodes is considered out-of-sync and should be updated with data from the other node. This state usually occurs during the recovery of a split-brain situation when the DRBD is manually invalidated.
- *Converted*: After converting nodes to a cluster (clicking **Convert to Cluster**) or enabling High Availability mode (clicking **Join HA**) and before rebooting the node(s).

Note
If you experience problems because the nodes of the HA cluster do not find each other during system startup, navigate to **Basic Settings > High Availability** and select **Make HA IP permanent**. That way the IP address of the HA interfaces of the nodes will be fix, which helps if the HA connection between the nodes is slow.

The **DRBD status** field indicates whether the latest data (including SCB configuration, audit trails, log files, and so on) is available on both SCB nodes. The master node (this node) must always be in **consistent** status to prevent data loss. Inconsistent status means that the data on the node is not up-to-date, and should be synchronized from the node having the latest data.

The **DRBD status** field also indicates the connection between the disk system of the SCB nodes. The following statuses are possible:

- *Connected*: Both nodes are functioning properly.
- *Invalidated*: The data on one of the nodes is considered out-of-sync and should be updated with data from the other node. This state usually occurs during the recovery of a split-brain situation when the DRBD is manually invalidated.
- *Sync source* or *Sync target*: One node (*Sync target*) is downloading data from the other node (*Sync source*).

Note
When the two nodes are synchronizing data, it is not possible to reboot or shutdown the master node. If you absolutely must shutdown SCB in such a situation, shutdown the slave node first, and then the master node.

When synchronizing data, the progress and the remaining time is displayed in the **System monitor**.

- *Split brain*: The two nodes lost the connection to each other, with the possibility of both nodes being active (master) for a time.

Warning
Hazard of data loss! In this case, valuable audit trails might be available on both SCB nodes, so special care must be taken to avoid data loss. For details on solving this problem, see *Procedure 11.7.2, Recovering from a split brain situation (p. 243).*

- *WFConnection*: One node is waiting for the other node; the connection between the nodes has not been established yet.

4.5.2.2. Adjusting the synchronization speed of DRBD

When operating two SCB units in High Availability mode, every incoming data copied from the master (active) node to the slave (passive) node. Since synchronizing data can take up significant system-resources, the maximal speed of the synchronization is limited, by default, to `10 MB/sec`. However, this means that synchronizing large amount of data can take very long time, so it is useful to increase the synchronization speed in certain situations — for example, when synchronizing the disks after converting a single node to a high availability cluster.

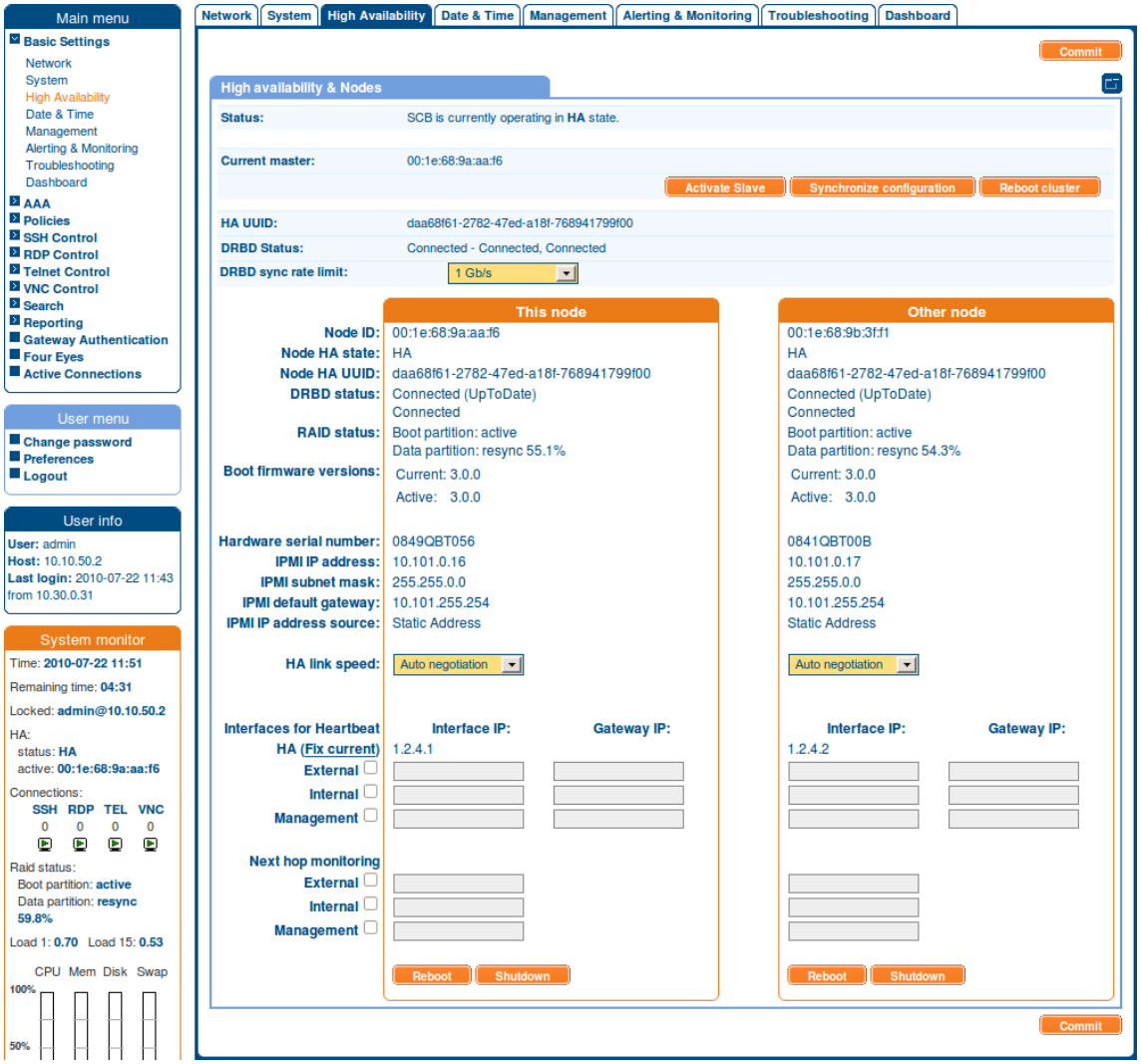


Figure 4.38. Adjusting DRBD synchronization speed

To change the limit of the DRBD synchronization rate, navigate to **Basic Settings > High Availability**, select **DRBD sync rate limit**, and select the desired value.

Note
Setting the sync rate to a high value is not recommended if the load of SCB is very high, because increasing the resources used by the synchronization process may degrade the general performance of SCB.

4.5.2.3. Procedure – Configuring redundant Heartbeat interfaces

Purpose:

In order to avoid unnecessary takeovers and minimize the chance of split-brain situations, it is possible to configure additional Heartbeat interfaces in SCB. These redundant Heartbeat interfaces are used only to detect that the other node is still available; it is not used to synchronize data between the nodes (only Heartbeat messages are transferred).

For example, if the main HA interface breaks down, or is accidentally unplugged and the nodes can still access each other on the redundant HA interface, no takeover occurs, but no data is synchronized to the slave node until the main HA link is restored. Similarly, if connection on the redundant Heartbeat interface is lost, but the main HA connection is available, no takeover occurs.

The redundant Heartbeat interface is a virtual interface that uses an existing interface of the SCB device (for example the external or the management interface). The original MAC address of the interface is displayed at **Basic Settings > High Availability > Interfaces for Heartbeat > Production MAC**, while the MAC address of the virtual redundant Heartbeat interface is displayed at **Basic Settings > High Availability > Interfaces for Heartbeat > HA MAC**. The MAC address of the redundant Heartbeat interface is generated in a way that it cannot interfere with the MAC addresses of physical interfaces.

Warning
In case the nodes lose connection on the main HA interface, and after a time the connection is lost on the redundant Heartbeat interfaces as well, the slave node will become active. However, as the master node was active for a time when no data synchronization was possible between the nodes, this results in a split-brain situation which must be resolved before the HA functionality can be restored. For details, see *Procedure 11.7.2, Recovering from a split brain situation (p. 243).*

To configure a redundant Heartbeat interface, complete the following steps.

Steps:

Step 1. Navigate to **Basic Settings > High Availability > Interfaces for Heartbeat**.

Step 2. Select the interface you want to use as redundant Heartbeat interface (for example `External`). Using an interface as a redundant Heartbeat interface does not affect the original traffic of the interface.

Warning
When using SCB in Bridge mode, use the `Management` interface for redundant HA functionality. Enabling the redundant HA option on the `External` or `Internal` interface in Bridge mode can cause network errors depending on your network architecture

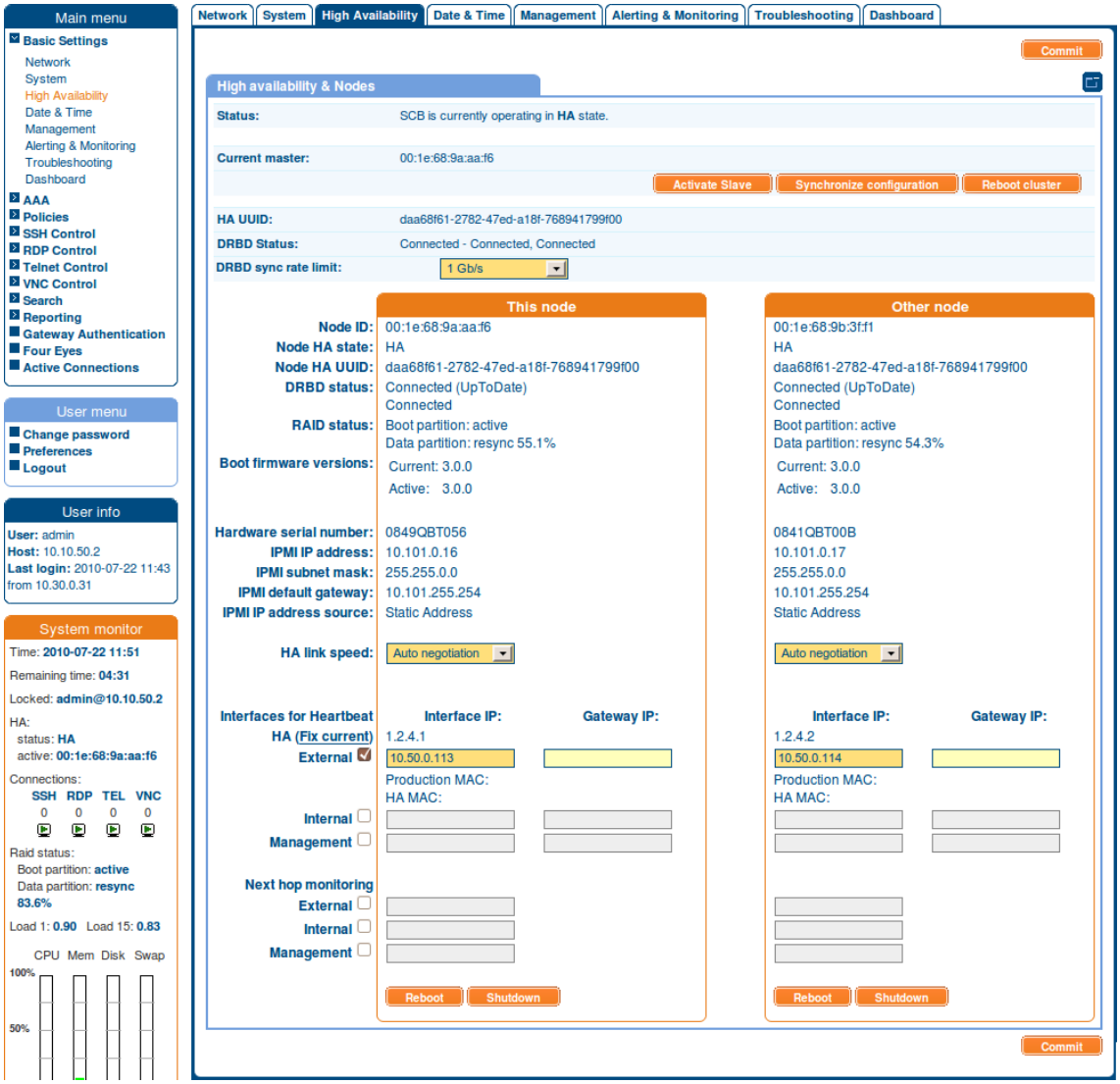


Figure 4.39. Configuring redundant Heartbeat interfaces

Step 3. Enter an IP address into the **This node > Interface IP** field of the selected interface. This IP address must be a real IP address that is visible from the other node. The two nodes cannot have the same IP address on their redundant Heartbeat interfaces. Enter the IP address of the gateway into the **Gateway IP** if needed.

Step 4. Enter an IP address into the **Other node > Interface IP** field of the selected interface. This IP address must be a real IP address that is visible from the other node. The two nodes cannot have the same IP address on their redundant Heartbeat interfaces. Enter the IP address of the gateway into the **Gateway IP** if needed.

Step 5. Repeat the previous steps to add additional redundant Heartbeat interfaces if needed.

Step 6. Click **Commit**.

Warning
For the changes to take effect, you have to restart both nodes. To restart both nodes, click **Reboot Cluster**.

4.5.2.4. Redundant Heartbeat interface status explained

The status of the redundant Heartbeat interfaces is displayed at **Basic Settings > High Availability > Redundant Heartbeat status**, and also in the **HA > Redundant** field of the System monitor. The possible status messages are explained below:

- *NOT USED*: There are no redundant Heartbeat interfaces configured.
- *OK*: Normal operation, every redundant Heartbeat interface is working properly.
- *DEGRADED-WORKING*: Two or more redundant Heartbeat interfaces are configured, and at least one of them is functioning properly. This status is displayed also when a new redundant Heartbeat interface has been configured, but the nodes of the SCB cluster has not been restarted yet.
- *DEGRADED*: The connection between the redundant Heartbeat interfaces has been lost. Investigate the problem to restore the connection.
- *INVALID*: An error occurred with the redundant Heartbeat interfaces. Contact the BalaBit Support Team for help. For contact details, see *Section 5, Contact and support information (p. xiv)*.

4.5.2.5. Procedure – Configuring next-hop router monitoring

Purpose:

By default, HA takeover occurs only if the master node stops working or becomes unreachable from the slave node. However, this does not cover the scenario when the master node becomes unaccessible to the outside world (for example its external interface or the router or switch connected to the external interface breaks down) while the slave node would be still accessible (for example because it is connected to a different router).

To address such situations, it is possible to specify IP addresses (usually next hop routers) which are continuously monitored from both the master and the slave nodes using ICMP echo (ping) messages. One such address can be set for every interface; the specified IP addresses must be on the same network segment as the respective interface of SCB. If any of the monitored addresses becomes unreachable from the master node while being reachable from the slave node (in other words, more monitored addresses are accessible from the slave node) than it is assumed that the master node is unreachable and a forced takeover occurs — even if the master node is otherwise functional.

Naturally, if the slave node is not capable of taking over the master node (for example because there is data not yet synchronized from the current master node) no takeover is performed.

To configure a next hop monitoring, complete the following steps.

Steps:

Step 1. Navigate to **Basic Settings > High Availability > Next hop monitoring**.

Step 2. Select the interface to use for monitoring its next-hop router.

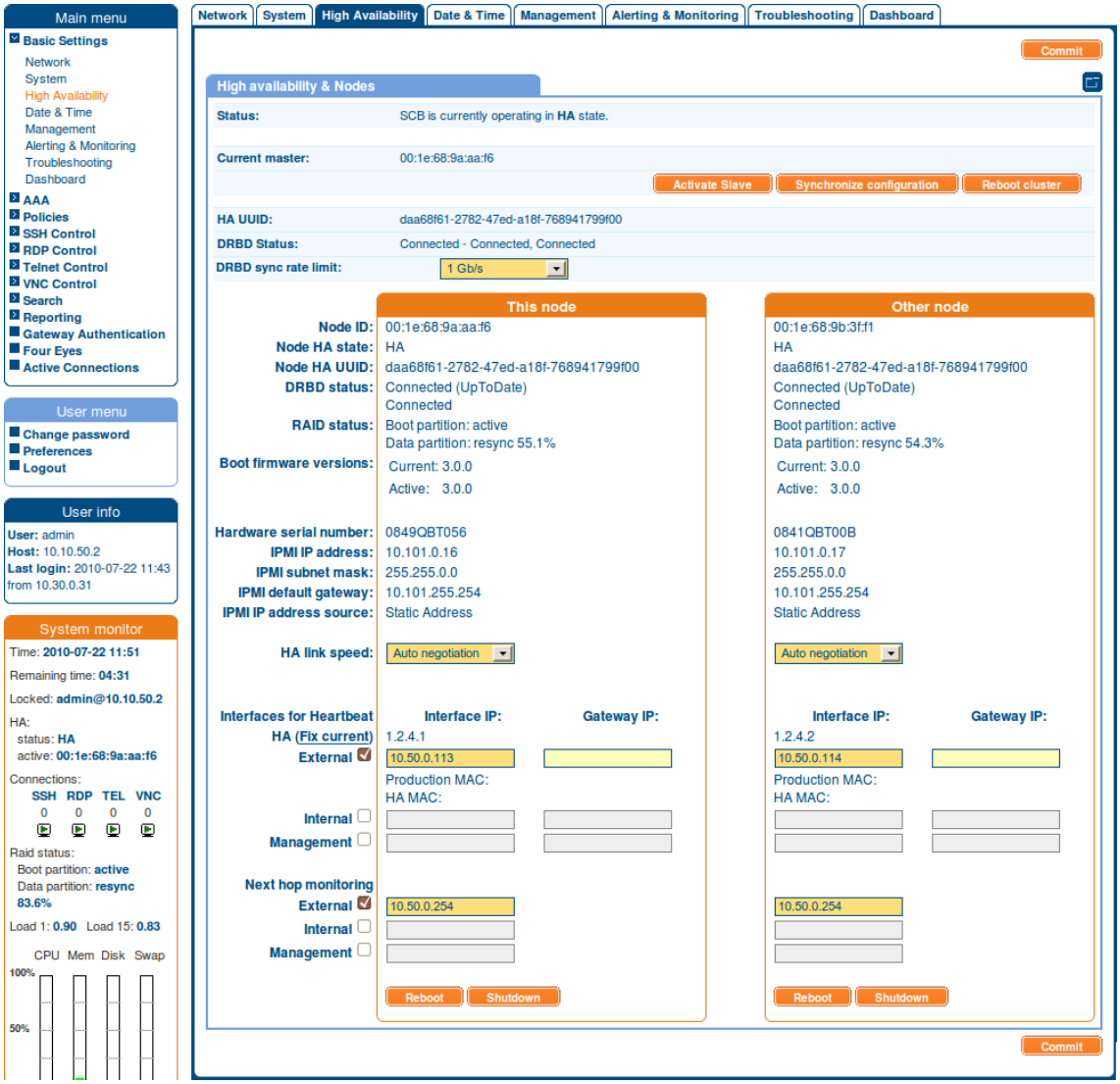


Figure 4.40. Configuring next hop monitoring

Step 3. Enter the IP address to monitor from the current master node (for example the IP address of the router or the switch connected to the interface) into the **This node > Next hop IP** field of the selected interface. This IP address must be a real IP address that is visible from the interface, and must be on the same local network segment.

Step 4. Enter the IP address to monitor from the current slave node (for example the IP address of the router or the switch connected to the interface) into the **Other node > Next hop IP** field of the selected interface. This IP address must be a real IP address that is visible from the interface, and must be on the same local network segment.

Step 5. Repeat the previous steps to add IP addresses to be monitored from the other interfaces if needed.

Step 6. Click **Commit**.

Warning
For the changes to take effect, you have to restart both nodes. To restart both nodes, click **Reboot Cluster**.

4.5.3. Upgrading SCB

The following sections describe how to keep SCB up to date, and how to install a new license file if needed.

- For details on how to upgrade SCB, see *Section 4.5.3.1, Updating the SCB firmware (p. 103)*.
- For details on how to upgrade the firmware of an SCB cluster, see *Procedure 4.5.3.1.2, Upgrading both the core and the boot firmware of a high availability system (p. 105)*.
- For details on how to install a new license file, see *Procedure 4.5.3.2, Updating the SCB license (p. 106)*.
- For details on how to import or export the configuration of SCB, see *Procedure 4.5.3.3, Exporting the configuration of SCB (p. 107)*.

Warning
Downgrading from a feature release to an earlier (and thus unsupported) feature release, or to the stable release is not supported: this means that once you upgrade a system from a stable release (for example 1.0) to a feature release (for example 1.1), you will have to keep upgrading to the new feature releases until the next stable version release (for example 2.0) is published, or risk using an unsupported product.

4.5.3.1. Updating the SCB firmware

SCB can be updated when a new firmware version is available. To display information about the firmware currently running on SCB, navigate to **Basic Settings > System > Version details**. The following is displayed:

- *Core firmware | Boot firmware*: The version number of the firmwares currently running on SCB (for example `2.0`).
- *Build date*: The date when the currently running firmware was created.

Updating SCB is described in the following sections.

Warning
Before uploading a new firmware image, backup the configuration of SCB. For details, see *Procedure 4.5.3.3, Exporting the configuration of SCB (p. 107)*.

Always read the release notes of the firmware before updating SCB, because the release notes may include special instructions specific to the firmware version. *The release notes are available here* .

- For details on how to upgrade a standalone SCB unit, complete *Procedure 4.5.3.1.1, Updating SCB and managing the firmware (p. 104)*.
- For details on how to upgrade the firmware of an SCB cluster, complete *Procedure 4.5.3.1.2, Upgrading both the core and the boot firmware of a high availability system (p. 105)*.

- For details on how to revert to an older firmware version (if possible), see *Procedure 4.5.3.1.3, Reverting to an older firmware version (p. 106).*

4.5.3.1.1. Procedure – Updating SCB and managing the firmware

Steps:

Step 1. Visit the BalaBit website and *download the latest firmware here* .

Step 2. Navigate to the **Basic Settings > System** page.

Step 3. To update the internal (core) firmware, select **Core firmwares**.
To update the external (boot) firmware, select **Boot firmwares**.

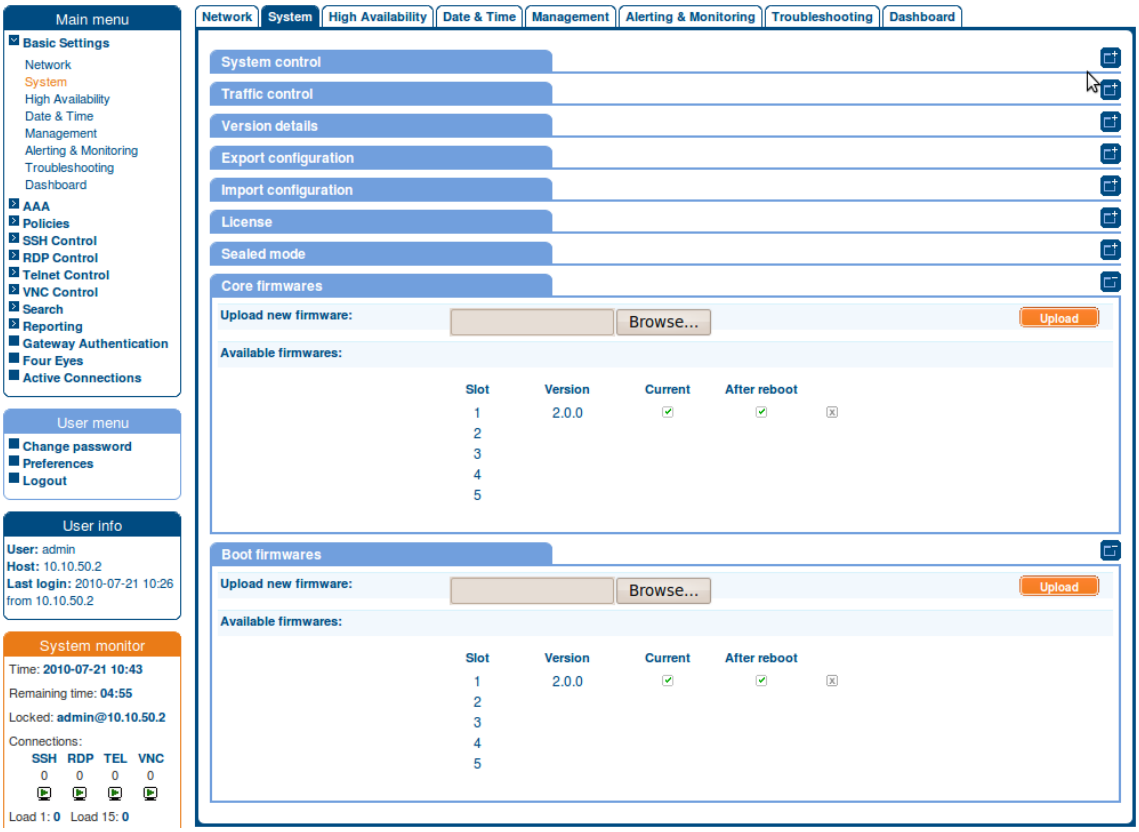


Figure 4.41. Managing the firmwares

For details on the different firmwares, see *Section 2.11, Firmware in SCB (p. 20).*

Step 4. Select the firmware file using the **Browse** button. The extension of firmware files is `.bin`

Step 5. Click **Upload**. After uploading, the new firmware is added to the **Available firmwares** list.

Step 6. Click ☐ in the **After reboot** column of the new firmware.

Step 7. Navigate to **System Control > This node** and click **Reboot**.

Tip
When SCB boots, it sends a message into the system log that includes the version numbers of both booted firmwares.

Note
If you experience any problems on the BalaBit Shell Control Box web interface after performing the upgrade, first empty the cache of your browser, or click the Reload button of your browser while holding the Shift key.

4.5.3.1.2. Procedure – Upgrading both the core and the boot firmware of a high availability system

Purpose:

If an SCB release requires the upgrading of both the boot firmware and the core (internal) firmware, complete the following steps:

Steps:

Step 1. Download both the core (internal) and the boot (external) firmware.

Step 2. Update the core firmware of the SCB using the web interface.

Step a. Navigate to **Basic Settings > System > Core firmwares** and upload the new core firmware.

Step b. When the upload is finished, select the **After reboot** option for the new firmware.

Warning
DO NOT REBOOT SCB AFTER UPGRADING THE CORE FIRMWARE.

Step 3. Repeat the previous step with the Boot firmware.

Step 4. Select **Basic Settings > High Availability > Reboot Cluster** to restart both nodes.

Warning
Hazard of data loss! As this step terminates all active connections, perform it only during maintenance hours to prevent data loss.

Note
If you experience any problems on the BalaBit Shell Control Box web interface after performing the upgrade, empty the cache of your browser, or click the Reload button of your browser while holding the Shift key.

4.5.3.1.3. Procedure – Reverting to an older firmware version

Purpose:

SCB can store up to five different firmware versions, any of them can be booted if required. The available firmwares are displayed on the **Basic Settings > System > Boot firmware** and **Basic Settings > System > Core firmware** pages. The list shows the detailed version of each firmware, including the version number, the revision number, and the build date. The firmware running on SCB is marked with in the **Current** column. The firmware that will be run after the next SCB reboot is marked with in the **After reboot** column.

To boot an older firmware, complete the following steps:

Warning
When upgrading SCB, it is possible that the configuration file is updated as well. In such cases, simply rebooting with the old firmware will not result in a complete downgrade, because the old firmware may not be able to read the new configuration file. If this happens, access the console menu of SCB, and select the **Revert Configuration** option to restore the configuration file to its state before the firmware was upgraded. For details on using the console menu, see *Section 4.5.4.1, Using the console menu of SCB (p. 110).*

Steps:

Step 1. Navigate to **Basic Settings > System > Boot firmware**.

Step 2. Select the firmware version to use, and click in the **After reboot** column.

Step 3. Navigate to **Basic Settings > System > Core firmware**.

Step 4. Select the firmware version to use, and click in the **Boot** column.

Step 5. Select **System Control > This node > Reboot** to reboot SCB.
If you are running an SCB cluster, select **Basic Settings > High Availability > Reboot Cluster**.

4.5.3.2. Procedure – Updating the SCB license

Purpose:

The SCB license must be updated before the existing license expires or when you purchase a new license. Information of the current license of SCB is displayed on the **Basic Settings > System > License** page. The following information is displayed:

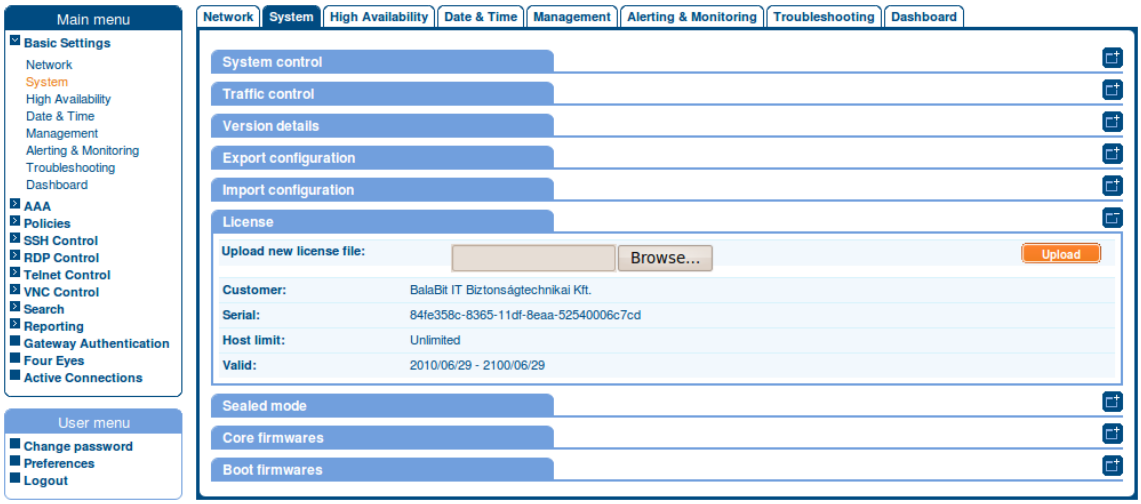


Figure 4.42. Updating the license

- *Customer*: The company permitted to use the license (for example `Example Ltd.`).
- *Serial*: The unique serial number of the license.
- *Host limit*: The number of servers that can be connected through SCB (for example `25`).

SCB gives an automatic alert one week before the license expires. An alert is sent also when the number of protected servers exceeds 120% of the number of servers set in the `Host limit` parameter of the license.

To update the license, complete the following steps:

Warning
Before uploading a new license, you are recommended to backup the configuration of SCB. For details, see *Procedure 4.5.3.3, Exporting the configuration of SCB (p. 107)*.

Steps:

Step 1. Navigate to **Basic Settings > System > License**.

Step 2. Click **Browse** and select the new license file.

Step 3. Click **Upload**, then **Commit**.

Step 4.

Warning
This step terminates all controlled connections going through SCB. Disconnect your clients from the protected servers before proceeding.

To activate the new license, navigate to **System control > All traffic** and click **Restart**.

4.5.3.3. Procedure – Exporting the configuration of SCB

Purpose:

The configuration of SCB can be exported (for manual archiving, or to migrate it to another SCB unit) from the **Basic Settings > System** page. Use the respective action buttons to perform the desired operation.

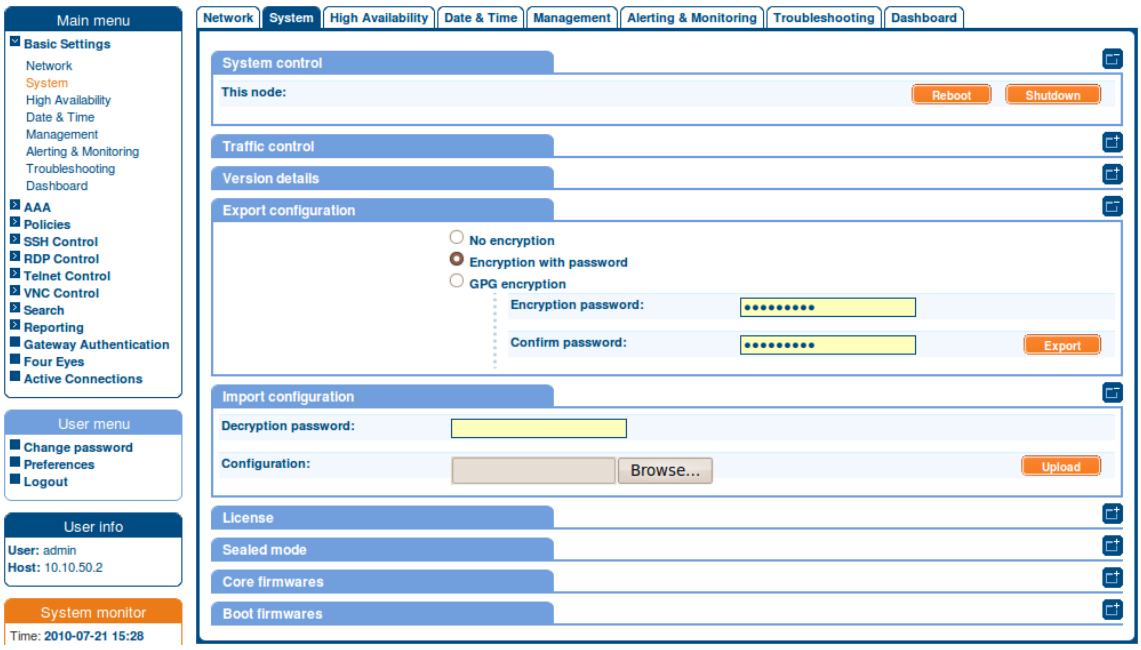


Figure 4.43. Exporting the SCB configuration

Steps:

Step 1. Navigate to **Basic Settings > System > Export configuration**.

Step 2. Select how to encrypt the configuration:

- To export the configuration file without encryption, select **No encryption**.

Warning
Exporting the SCB configuration without encyption is not recommended, as it contains sensitive information such as password hashes and private keys.

- To encrypt the configuration file with a simple password, select **Encrypt with password** and enter the password into the **Encryption password** and **Confirm password** fields.
- To encrypt the configuration file with GPG, select **GPG encryption**. Note that this option uses the same GPG key that is used to encrypt automatic system backups, and is only available if you have uploaded the public part of a GPG key to SCB at **Basic > System > Management > System backup**. For details, see *Procedure 4.3.5.3, Encrypting configuration backups with GPG (p. 69)*.

Step 3. Click **Export**.

Note
The exported file is a gzip-compressed archive. On Windows platforms, it can be decompressed with common archive managers such as the *free 7-Zip tool*.

The name of the exported file is `<hostname_of_SCB>-YYYMMDDTHHMM.config`; the `-encrypted` or `-gpg` suffix is added for password-encrypted and GPG-encrypted files, respectively.

4.5.3.4. Procedure – Importing the configuration of SCB

Purpose:

The configuration of SCB can be imported from the **Basic Settings > System** page. Use the respective action buttons to perform the desired operation.

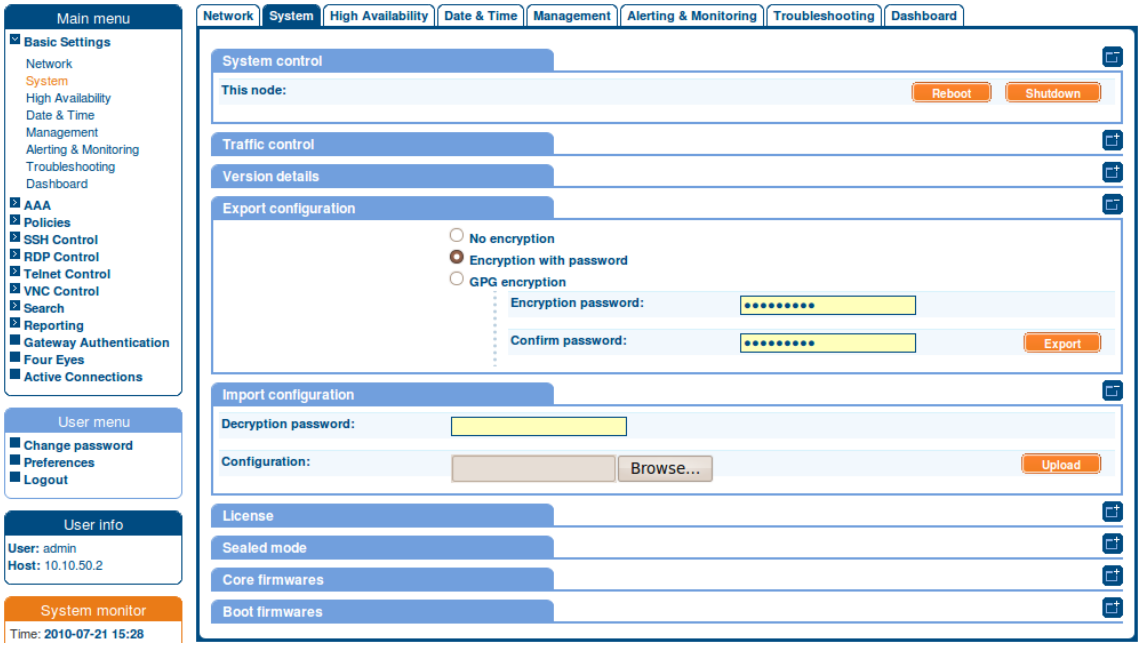


Figure 4.44. Importing the SCB configuration

Warning
It is not possible to import the configuration of an older major release (for example, 1.0) into a newer release (for example, 2.0).

Steps:

Step 1. Navigate to **Basic Settings > System > Import configuration**.

Step 2. Click **Browse** and select the configuration file to import.

Step 3. Enter the password into the **Encryption password** field and click **Upload**.

4.5.4. Accessing the SCB console

This section describes how to use the console menu of SCB, how to enable remote SSH access to SCB, and how to change the root password from the web interface.

Tip
If you need to find the SCB appliance in the server room, navigate to **Basic Settings > System > Hardware informantion > Blink system identification lights** and click **On**. This will blink the LEDs of hard disk trays on the front of the SCB appliance in red. Note that this is available only for *BalaBit Shell Control Box N10000*.

4.5.4.1. Using the console menu of SCB

Connecting to the BalaBit Shell Control Box locally or remotely using Secure Shell (SSH) allows you to access the console menu of SCB. The console menu provides access to the most basic configuration and management settings of SCB. It is mainly used for troubleshooting purposes; the primary interface of SCB is the web interface.

The console menu is accessible to the `root` user using the password set during completing the Welcome Wizard.

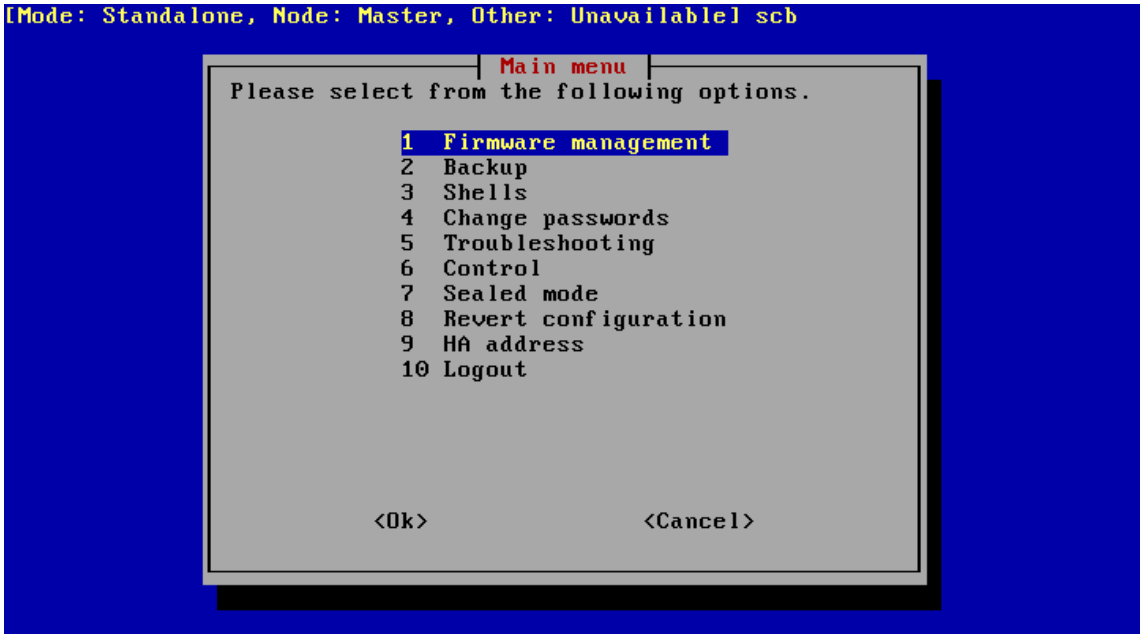


Figure 4.45. The console menu

The console menu provides allows you to perform the following actions:

- Select the active core and boot firmwares, and delete unneeded firmwares. Accessing the firmware management is useful if after an update the new firmware does not operate properly and the web interface is not available to activate the previous firmware.
- Start backup processes.
- Change the passwords of the `root` and `admin` users.

- Access the local shells of the core and boot firmwares. This is usually not recommended and only required in certain troubleshooting situations.
- Access the network-troubleshooting functions and display the available log files.
- Reboot and shutdown the system.
- Enable and disable sealed mode. For details, see *Section 4.5.5, Sealed mode (p. 113)*.
- Revert the configuration file. For details, see *Procedure 4.5.3.1.3, Reverting to an older firmware version (p. 106)*.
- Set the IP address of the HA interface.

Note
Note that logging in to the console menu automatically locks the SCB interface, meaning that users cannot access the web interface while the console menu is used. The console menu can be accessed only if there are no users accessing the web interface. The connection of web-interface users can be terminated to force access to the console menu.

4.5.4.2. Procedure – Enabling SSH access to the SCB host

Purpose:

Exclusively for troubleshooting purposes, you can access the SCB host using SSH. Completing the Welcome Wizard automatically disables SSH access. To enable it again, complete the following steps:

Warning
Accessing the SCB host directly using SSH is not recommended nor supported, except for troubleshooting purposes. In such case, the BalaBit Support Team will give you exact instructions on what to do to solve the problem.

Enabling the SSH server allows you to connect remotely to the SCB host and login using the `root` user. The password of the root user is the one you had to provide in the Welcome wizard. For details on how to change the root password from the web interface, see *Procedure 4.5.4.3, Changing the root password of SCB (p. 112)*

Steps:

Step 1. Navigate to **Basic Settings > Management > SSH settings**.

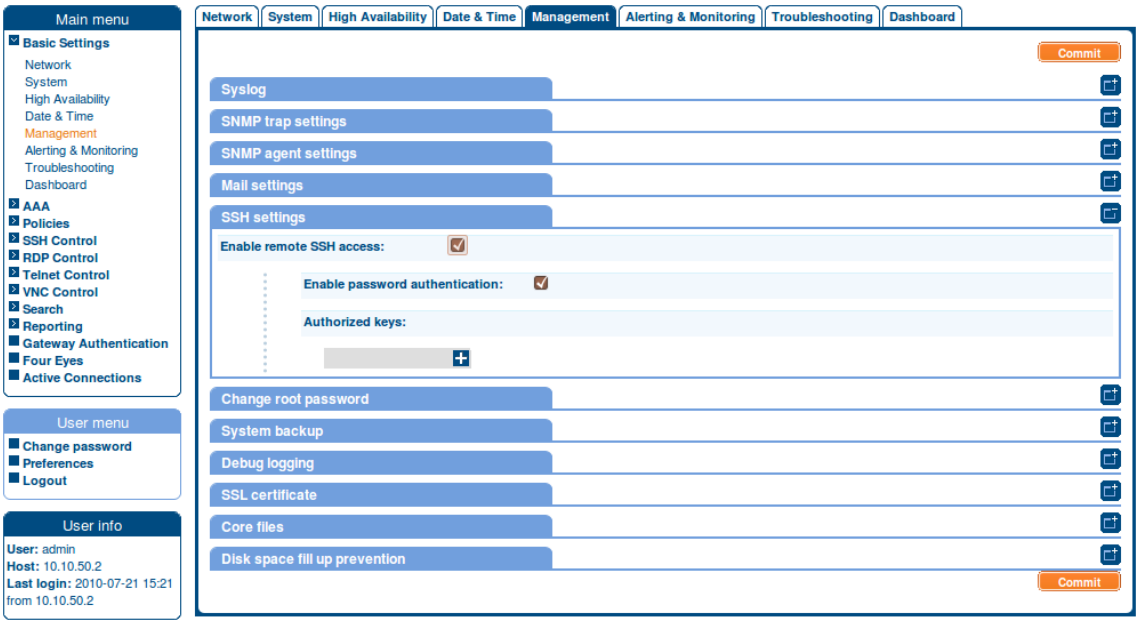


Figure 4.46. Enabling remote SSH access to SCB

Step 2. Select the **Enable remote SSH access** option.

Note
Remote SSH access is automatically disabled if Sealed mode is enabled. For details, see *Section 4.5.5, Sealed mode (p. 113)*.

Step 3. Set the authentication method for the remote SSH connections.

- To enable password-based authentication, select the **Enable password authentication** option.
- To enable public-key authentication, click ⊞ in the **Authorized keys** field, click ✎ and upload the private keys of the users who can access and manage SCB remotely via SSH.

Step 4. Click **Commit**.

The SSH server of SCB accepts connections only on the management interface if the management interface is configured. If the management interface is not configured, the SSH server accepts connections on the external interface. If possible, avoid enabling the SSH server of SCB when the management interface is not configured. For details on enabling the management connection, see *Procedure 4.3.1.1, Configuring the management interface (p. 47)*.

If you use SCB in Bastion mode and the management interface is not configured, read *Section 9.2.3, Accessing the SCB host in Bastion mode using SSH (p. 224)* for details on accessing the SCB host using SSH.

4.5.4.3. Procedure – Changing the root password of SCB

Purpose:

The root password is required to access SCB locally, or remotely via an SSH connection. Note that the password of the `root` user can be changed from the console menu as well. For details, see *Section 4.5.4, Accessing the SCB console (p. 110)*.

Steps:

Step 1. Navigate to **Basic Settings > Management > Change root password**.

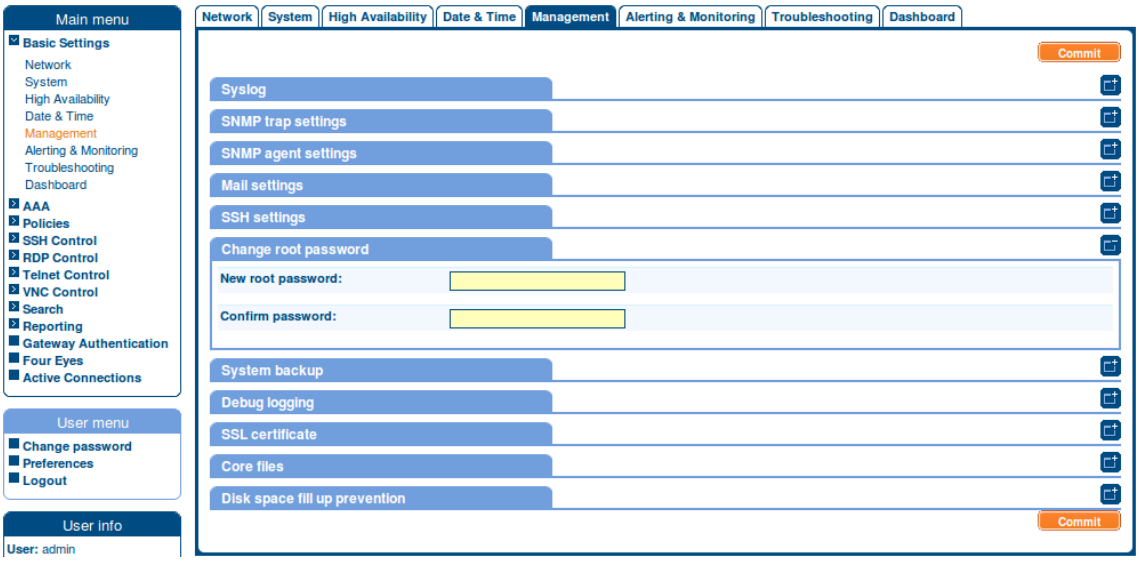


Figure 4.47. Changing the root password of SCB

Step 2. Enter the new password into the **New root password** and **Confirm password** fields.

Note
SCB passwords can contain the following special characters: `!"#$%&'()*+,-./:;<=>?@[\]^_`{|}`

Step 3. Click **Commit**.

4.5.5. Sealed mode

When sealed mode is enabled, the following settings are automatically applied:

- SCB cannot be accessed remotely via SSH for maintenance;
- the root password of SCB cannot be changed in sealed mode;
- Sealed mode can be disabled only from the local console. For details, see *Procedure 4.5.5.1, Disabling sealed mode (p. 114)*.

To enable sealed mode use one of the following methods:

- Select the **Sealed mode** option during the Welcome Wizard.

- Select **Basic Settings > System > Sealed mode > Activate sealed mode** on the SCB web interface.
- Login to SCB as root using SSH or the local console, and select **Sealed mode > Enable** from the console menu.

4.5.5.1. Procedure – Disabling sealed mode

Purpose:

To disable sealed mode, complete the following steps:

Steps:

Step 1. Go to the SCB appliance and access the local console.

Step 2. Login as `root`.

Step 3. From the console menu, select **Sealed mode > Disable**

Step 4. Select **Back to Main menu > Logout**.

4.5.6. Out-of-band management of SCB

SCB 3 LTS includes a dedicated out-of-band management interface conforming to the Intelligent Platform Management Interface (IPMI) v2.0 standards. The IPMI interface allows system administrators to monitor the system health of SCB and to manage the computer events remotely, independently of the operating system of SCB. SCB is accessible using the IPMI interface only if the IPMI interface is physically connected to the network.

- For details on connecting the IPMI interface, see *Procedure 3.1, Installing the SCB hardware (p. 250).*
- For details on configuring and using the IPMI interface to remotely monitor and manage SCB, see the following document:
  The *Onboard BMC/IPMI User's Guide*, available at *the BalaBit Hardware Documentation page*.

Basic information about the IPMI interface is available also on the SCB web interface on the **Basic Settings > High Availability** page. The following information is displayed:

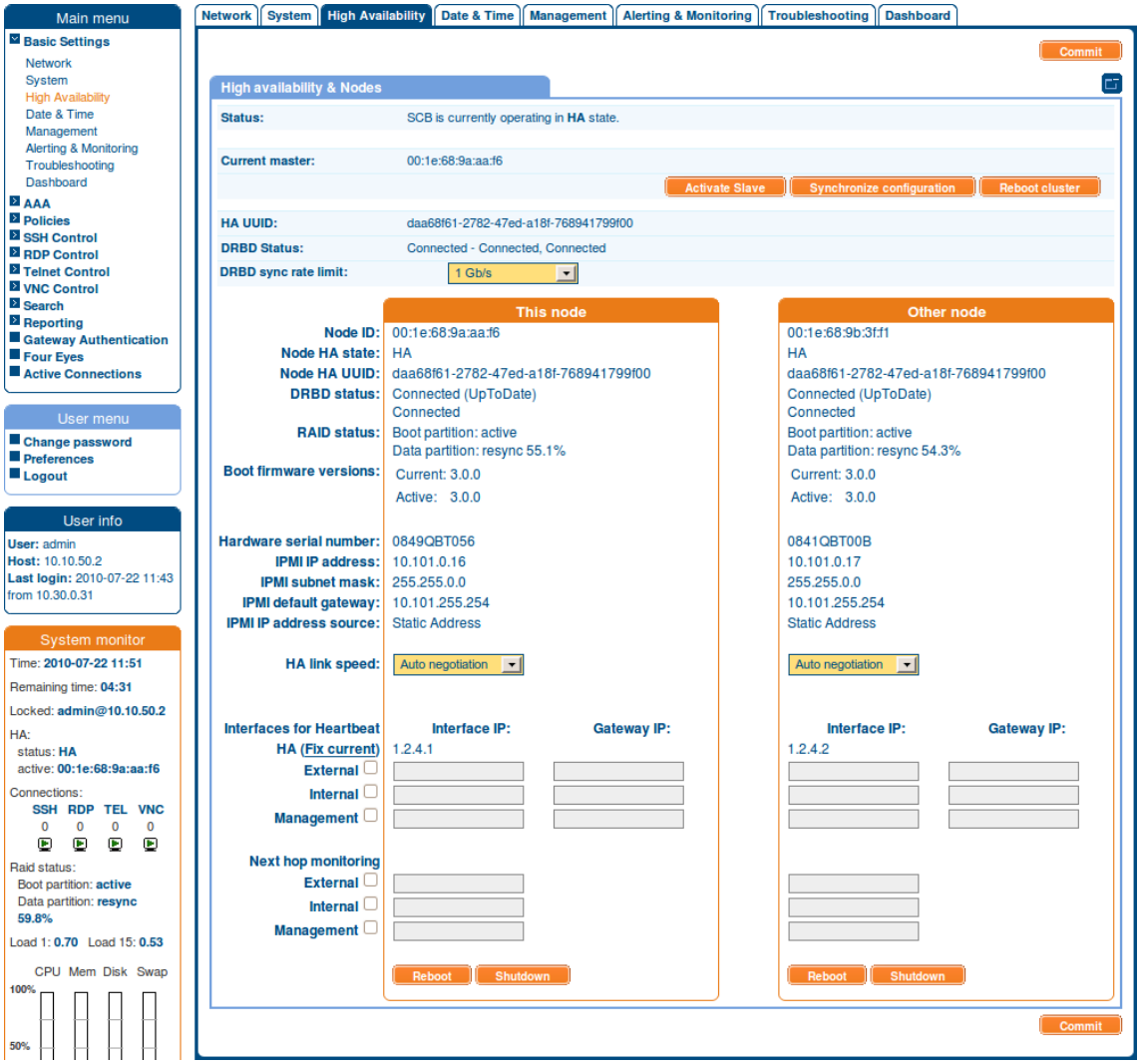


Figure 4.48. Information about the IPMI interface SCB

IPMI default gateway IP: The address of the default gateway configured for the IPMI interface.

- *Hardware serial number*: The unique serial number of the appliance.
- *IPMI IP address*: The IP address of the IPMI interface.
- *IPMI subnet mask*: The subnet mask of the IPMI interface.
- *IPMI IP address source*: Shows how the IPMI interface receives its IP address: dynamically from a DHCP server, or it uses a fixed static address.

Tip
If you need to find the SCB appliance in the server room, navigate to **Basic Settings > System > Hardware informantion > Blink system identification lights** and click **On**. This will blink the LEDs of hard disk trays on the front of the SCB appliance in red. Note that this is available only for *BalaBit Shell Control Box N10000*.

4.5.7. Managing the certificates used on SCB

SCB uses a number of certificates for different tasks that can be managed from the **Basic Settings > Management > SSL certificate** menu.

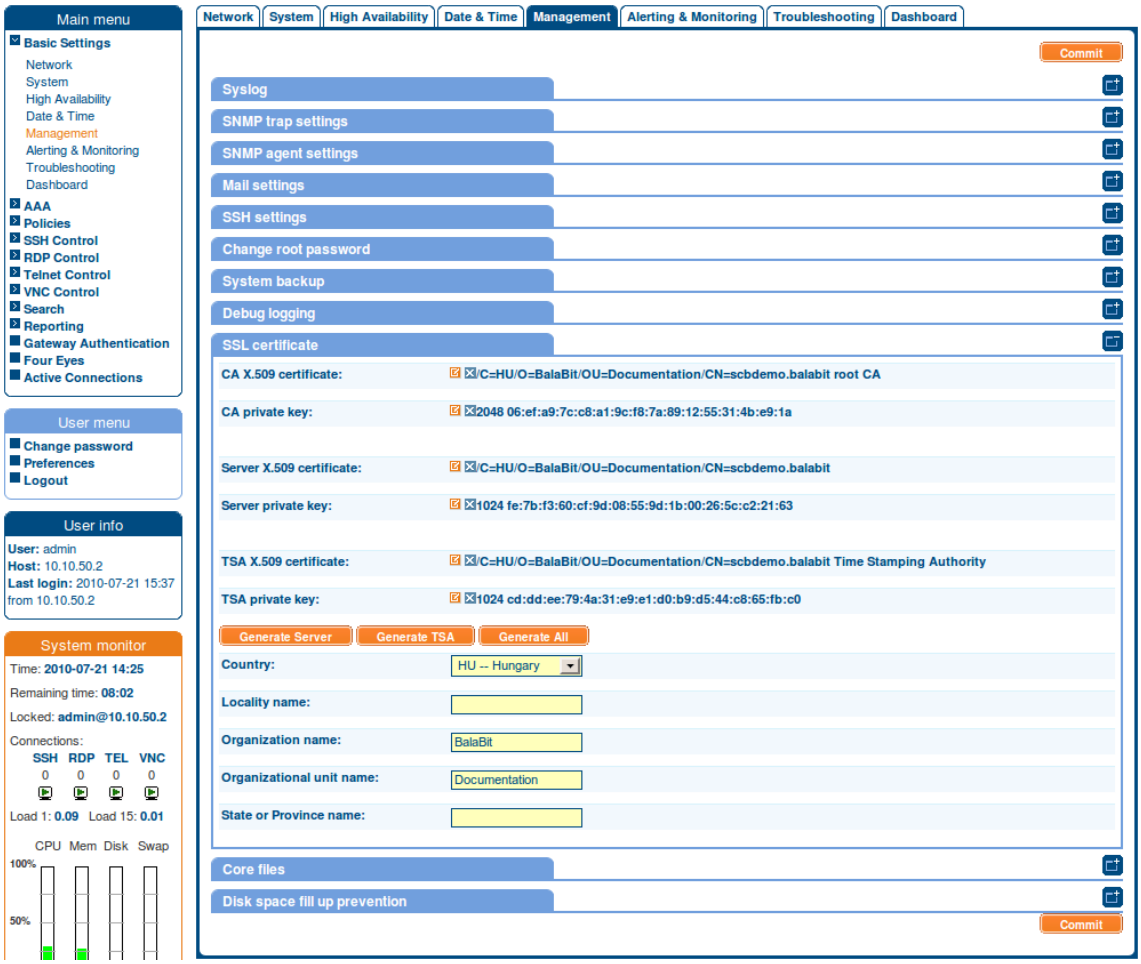


Figure 4.49. Changing the web certificate of SCB

The following certificates can be modified here:

- *CA certificate*: The certificate of the internal Certificate Authority of SCB.
- *Server certificate*: The certificate of the SCB web interface, used to encrypt the communication between SCB and the administrators.

Note
If this certificate is changed, the browser of SCB users will display a warning stating that the certificate of the site has changed.

- *TSA certificate*: The certificate of the internal Timestamping Authority that provides the timestamps used when creating encrypted audit-trails.

For every certificate, the distinguished name (DN) of the X.509 certificate and the fingerprint of the private key is displayed. To display the entire certificate click on the DN; to display the public part of the private key, click on the fingerprint. It is not possible to download the private key itself from the SCB web interface, but the certificate can be downloaded in different formats (for example PEM, OpenSSH, Tectia).

Note
Other parts of SCB may use additional certificates that are not managed here.

During the initial configuration, SCB creates a self-signed CA certificate, and uses this CA to issue the certificate of the web interface (see *Server certificate*) and the internal Timestamping Authority (*TSA certificate*).

There are two methods to manage certificates of SCB:

- *Recommended*: Generate certificates using your own PKI solution and upload them to SCB.
  Generate a CA certificate and two other certificates signed with this CA using your PKI solution and upload them to SCB. For the Server and TSA certificates, upload the private key as well.

  For details on uploading certificates and keys created with an external PKI, complete *Procedure 4.5.7.2, Uploading external certificates to SCB (p. 118)*.

Warning
The Server and the TSA certificates must be issued by the same Certificate Authority.

- Use the certificates generated on SCB. In case you want to generate new certificates and keys for SCB using its self-signed CA certificate, or generate a new self-signed CA certificate, complete *Procedure 4.5.7.1, Generating certificates for SCB (p. 117)*.

Note
Generate certificates using your own PKI solution and upload them to SCB whenever possible. Certificates generated on SCB cannot be revoked, and can become a security risk if they are somehow compromised.

4.5.7.1. Procedure – Generating certificates for SCB

Purpose:

Create a new certificate for the SCB webserver or the Timestamping Authority using the internal CA of SCB, or create a new, self-signed CA certificate for the internal Certificate Authority of SCB.

Steps:

Step 1. Navigate to **Basic Settings > Management > SSL certificate**.

Step 2. Fill the fields of the new certificate:

Step a. *Country*: Select the country where SCB is located (for example HU - Hungary).

Step b. *Locality*: The city where SCB is located (for example Budapest).

Step c. *Organization*: The company who owns SCB (for example Example Inc.).

Step d. *Organization unit*: The division of the company who owns SCB (for example IT Security Department).

Step e. *State or Province*: The state or province where SCB is located.

Step 3. Select the certificate you want to generate.

- To create a new certificate for the SCB web interface, select **Generate Server certificate**.
- To create a new certificate for the Timestamping Authority, select **Generate TSA certificate**.
- To create a new certificate for the internal Certificate Authority of SCB, select **Generate All**. Note that in this case new certificates are created automatically for the server and TSA certificates as well.

Note
When generating new certificates, the server and TSA certificates are signed using the certificate of the CA. If you have uploaded an external CA certificate along with its private key, it will be used to create the new server and TSA certificates. If you have uploaded an external CA certificate without its private key, use your external PKI solution to generate certificates and upload them to SCB.

Warning
Generating a new certificate automatically deletes the earlier certificate.

Step 4. Click **Commit**.

4.5.7.2. Procedure – Uploading external certificates to SCB

Purpose:

Upload a certificate generated by an external PKI system to SCB.

Prerequisites:

The certificate to upload. For the TSA and Server certificate, the private key of the certificate is needed as well. The certificates must meet the following requirements:

- SCB accepts certificates in PEM format. The DER format is currently not supported.
- SCB accepts private keys in PEM (RSA and DSA), PUTTY, and SSHCOM/Tectia format. Password-protected private keys are also supported.

  For the internal CA certificate of SCB, uploading the private key is not required.

- For the TSA certificate, the *X509v3 Extended Key Usage* attribute must be enabled and set to *critical*. Also, its default value must be set to *Time Stamping*.

- For the Server certificate, the *X509v3 Extended Key Usage* attribute must be enabled and its default value set to *TLS Web Server Authentication*. Also, the Common Name of the certificate must contain the domain name or the IP address of the SCB host.
- For the certificate used to sign audit trails, the *X509v3 Extended Key Usage* attribute must be enabled and its default value set to *TLS Web Server Authentication*.

Steps:

Step 1. Navigate to **Basic Settings > Management > SSL certificate**.

Step 2. To upload a new certificate, click next to the certificate you want to modify. A popup window is displayed.

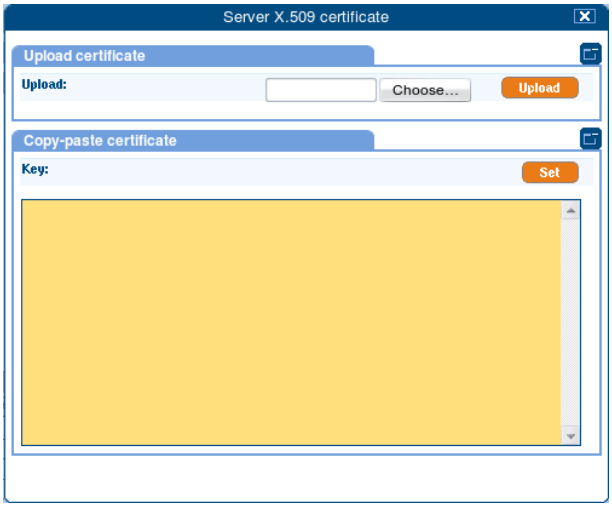


Figure 4.50. Uploading certificates

Select **Browse**, select the file containing the certificate, and click **Upload**. Alternatively, you can also copy-paste the certificate into the **Certificate** field and click **Set**.

Step 3. To upload the private key corresponding to the certificate, click the icon next to the private key you want to modify. A popup window is displayed.
Select **Browse**, select the file containing the certificate, and click **Upload**. Alternatively, you can also copy-paste the certificate into the **Key** field and click **Set**.

Chapter 5. Configuring connections

Connections determine if a server can be accessed from a particular client. The policies used in the connection definition can restrict the availability of the *connection* based on the username, time, authentication method, and so on. Channel policies (see *Procedure 5.1.4, Creating and editing channel policies (p. 126)*) determine if a particular channel can be used within an already established connection. The policies used in the channel policy can restrict the availability of the *channel* based on the server and the client IP address, username, and so on. The types of policies available in a connection depend on the protocol (SSH, RDP, and so on) enabled in the connection.

SCB supports the following protocols:

- Secure Shell version 2 (SSHv2)
- RDP versions 4-7
- Telnet and TN3270, as described by the relevant RFCs
- The Virtual Networking (VNC) protocol versions 3.3-3.8
- VMware View when VMware View Clients using the Remote Desktop (RDP) display protocol to access remote servers. For details, see *Section 5.6, VMware View connections (p. 173)*.

5.1. General connection settings

This section describes how to configure connections, and details the general configuration options and policies that apply to every type of connection that SCB can control: SSH, RDP, Telnet, and VNC.

Protocol-specific configuration options are described in their respective sections: *Section 5.2, SSH-specific settings (p. 148)*, *Section 5.3, RDP-specific settings (p. 165)*, *Section 5.4, Telnet-specific settings (p. 172)*, and *Section 5.5, VNC-specific settings (p. 172)*.

5.1.1. Procedure – Configuring connections

Purpose:

To configure a connection, complete the following steps.

Steps:

Step 1. Select the type of connection from the main menu.

- To configure a Secure Shell connection, select **SSH Control > Connections**.
- To configure a Remote Desktop connection, select **RDP Control > Connections**.
- To configure a Telnet connection, select **Telnet Control > Connections**.
- To configure a VNC connection, select **VNC Control > Connections**.

Step 2.

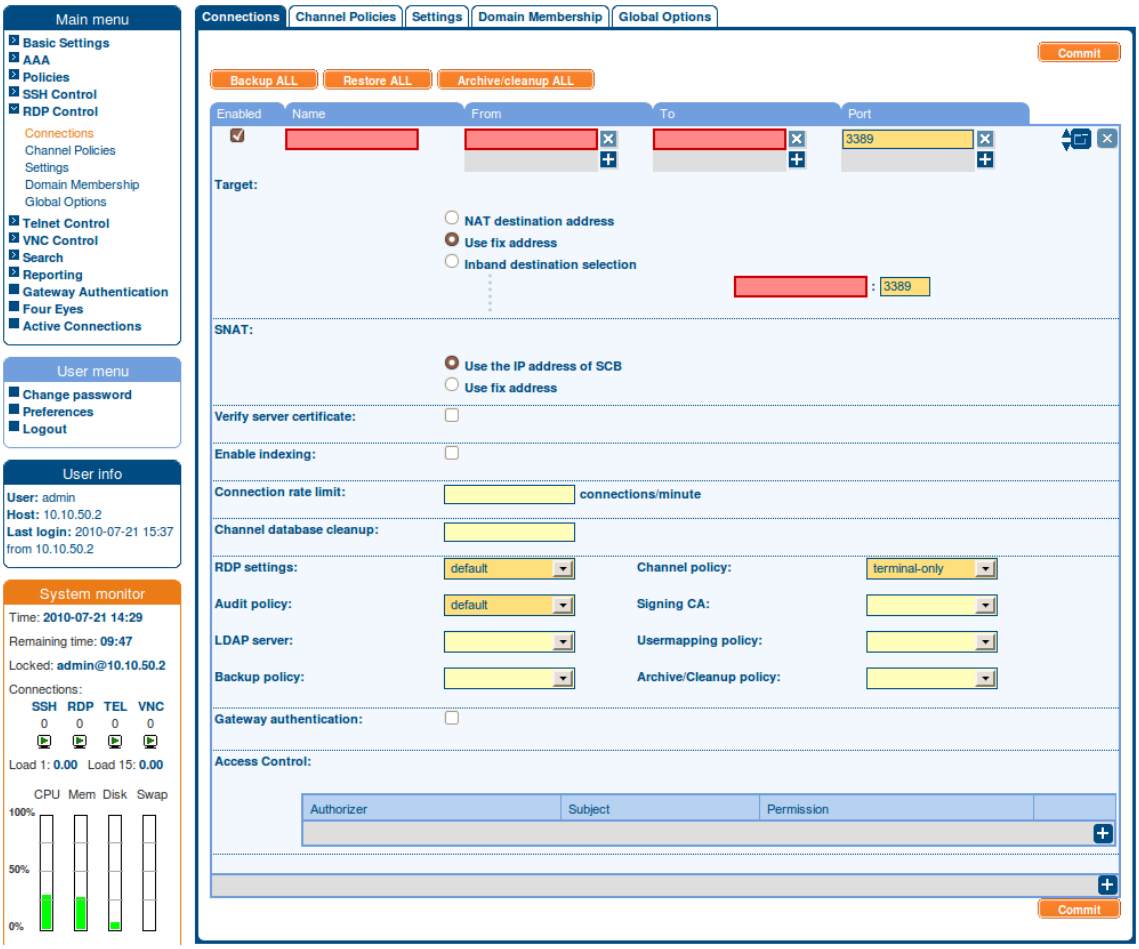


Figure 5.1. Configuring connections

Click ➕ to define a new connection and enter a name that will identify the connection (for example `admin_mainserver`).

Tip
It is recommended to use descriptive names that give information about the connection, for example refer to the name of the accessible server, the allowed clients, and so on.

Step 3. Enter the IP address of the client that will be permitted to access the server into the **From** field. Click ➕ to list additional clients.

Step 4. Enter the IP address that the clients will request into the **To** field.

- In Bastion mode, enter the IP address of SCB's external interface. If the audited connection is initiated on a protected server, enter the IP address of SCB's internal interface.
- In Bridge mode, enter the IP address of the protected server.
- In Router mode, enter the IP address of the protected server.

Click ⊞ to list additional IP addresses.

Step 5. If the clients use a custom port to address the server instead of the default port used by the protocol, enter the port number that the clients will request into the **Port** field. Click ⊞ to list additional port numbers.

Step 6. *Bastion mode*: Enter the IP address and port number of the target server into the **Target** field. SCB will connect all incoming client-side connections to this server. For details on organizing connections in Bastion mode, see *Section 9.2.3, Accessing the SCB host in Bastion mode using SSH (p. 224)*.

Step 7. Configure advanced settings if needed, like network address translation, channel policy, gateway authentication, various policies, or other settings.

Step 8. Click **Commit** to save the connection.

Tip
To temporarily disable a connection, deselect the checkbox before the name of the connection.

Step 9. Depending on your needs and environment, you may want to set further settings for your connections.

- To modify the destination or source addresses of the connections, see *Procedure 5.1.2, Modifying the destination address (p. 123)* and *Procedure 5.1.3, Modifying the source address (p. 125)*.
- Select a **Backup Policy** and an **Archiving Policy** for the audit trails and indexes of the connection. For details, see *Section 4.3.5, Data and configuration archiving and backups (p. 62)*.

Note
The backup and archive policies set for the connection operate only on the audit trails and indexes of the connection. General data about the connections that is displayed on the **Search** page is archived and backed up as part of the system-backup process of SCB.

- If you want to timestamp, encrypt, or sign the audit trails, configure and **Audit Policy** to suit your needs. For details, see *Section 5.1.8, Audit policies (p. 135)*.
- To require the users to authenticate themselves not only on the target server, but on SCB as well, see *Section 8.2, Configuring gateway authentication (p. 210)*.
- To require 4-eyes authorization on the connections, with the possibility of an auditor monitoring the connection in real-time, see *Section 8.3, Configuring 4-eyes authorization (p. 215)*.
- Certain connections and scenarios (for example SSH authentication, gateway authentication, RDP6 connections) SCB can authenticate the user to an LDAP database, or retrieve the group memberships of the user. To use these features, select an **LDAP Server**. For details, see *Procedure 5.1.7, Authenticating users to an LDAP server (p. 131)*.
- To limit the number of new connection requests accepted from a single client IP address per minute, enter the maximal number of accepted connections into the **Connection rate limit** field.

Note
Protocol-specific configuration options are described in their respective sections: *Section 5.2, SSH-specific settings (p. 148)*, *Section 5.3, RDP-specific settings (p. 165)*, *Section 5.4, Telnet-specific settings (p. 172)*, and *Section 5.5, VNC-specific settings (p. 172)*.

5.1.2. Procedure – Modifying the destination address

Purpose:

The destination address is the address of the server where the clients finally connect to. To modify the destination address of a connection, complete the following steps.

Steps:

Step 1. Navigate to the **Connections** tab storing the connection and click to display the details of the connection.

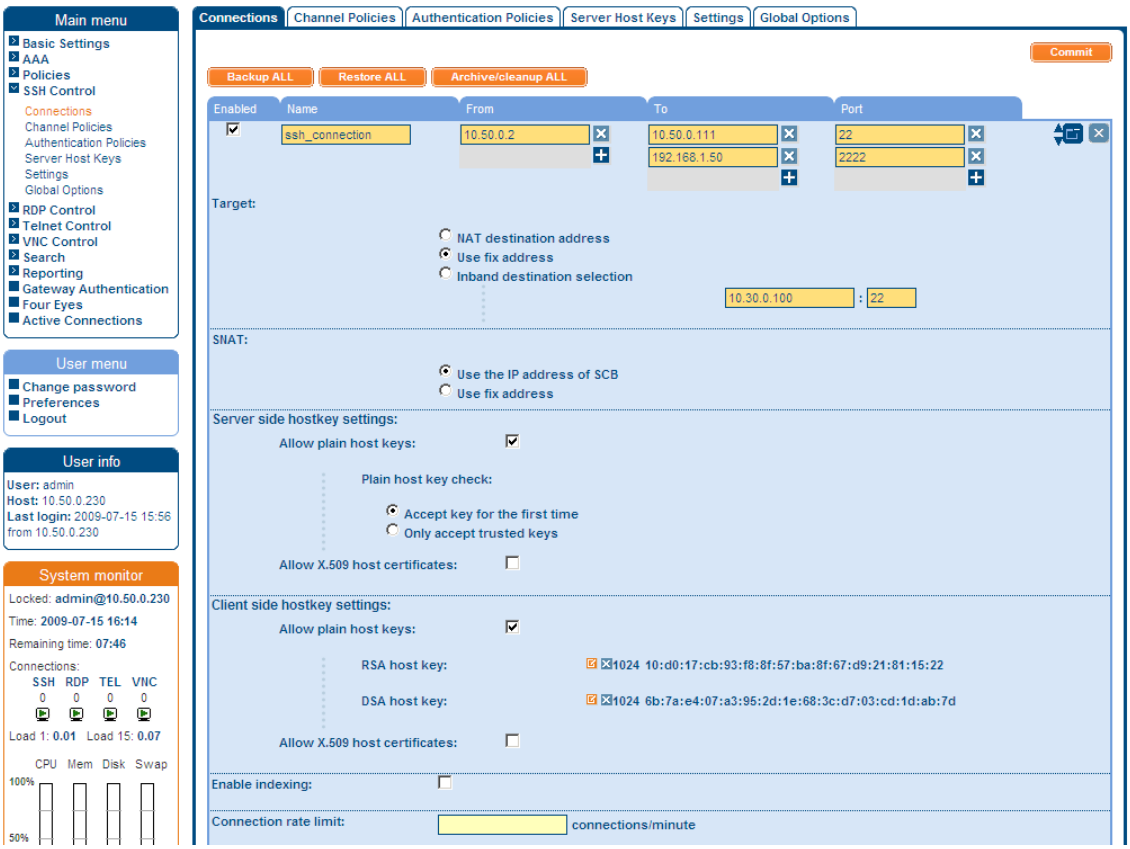


Figure 5.2. Configuring connections

Step 2. The **Target** section allows you to configure Network Address Translation (NAT) on the server side of SCB. Destination NAT determines the target IP address of the server-side connection. Set the destination address as required. The following options are available:

Note
It is not possible to direct the traffic to the IP addresses belonging to SCB.

- **Use the original target address of the client**: Connect to the IP address targeted by the client. This is the default behavior in Router and Bridge mode. This option is not available in Bastion mode.
- **NAT destination address**: Perform a network address translation on the target address. Enter the target address in IP address/Netmask format.
- **Use fix address**: Enter the IP address and port number of the server. The connection will connect always to this address, redirecting the clients to the server.
- **Inband destination selection**: Extract the address of the server from the username. This method is most commonly used in nontransparent Bastion mode (for a detailed example, see *Procedure 9.3, Using nontransparent Bastion mode (p. 224)*). Enter the IP address or the hostname of the domain name server used to resolve the address of the server into the **DNS Server** field.

  If the clients do not include the domain name when addressing the server (for example they use `username@server` instead of `username@server1.example.com`), SCB can automatically add domain information (for example `example.com`). Enter the domain name to add into the **Append domain** field.

Note
Inband destination selection is not available for Telnet and VNC connections.

SCB can also resolve CNAME records.

Note
If the **Append domain** field is filled, SCB will always add this suffix to the hostname. If you specify a DNS suffix, make sure that the clients do not include the domain name in their requests.

Enter the addresses of the servers that the users are permitted to access into the **Targets** field. Use the IP address/netmask (for example `192.168.2.16/32`) format, or enter the hostname of the server. The hostnames may contain the `*` and `?` wildcard characters. If the clients address the server using its IP address, make sure to include the IP address of the server in the Targets list. This is required because SCB resolves the hostnames to IP addresses, but does not reverse-resolve IP addresses to hostnames.

Warning
If only the hostname of the server is listed and the client addresses the server using its IP address, SCB will refuse the connection.

Leave the **Port** field empty if the clients may access any port on the server, or enter the specified port they can access.

If there are any servers that the users cannot target using inband destination selection, add them to the **Exceptions** field.

Step 3. Click **Commit**.

5.1.3. Procedure – Modifying the source address

Purpose:

The source address is the address that SCB uses to connect the server. The server sees this address as the source of the connection. To modify the source address of a connection, complete the following steps.

Steps:

Step 1. Navigate to the **Connections** tab storing the connection and click to display the details of the connection.

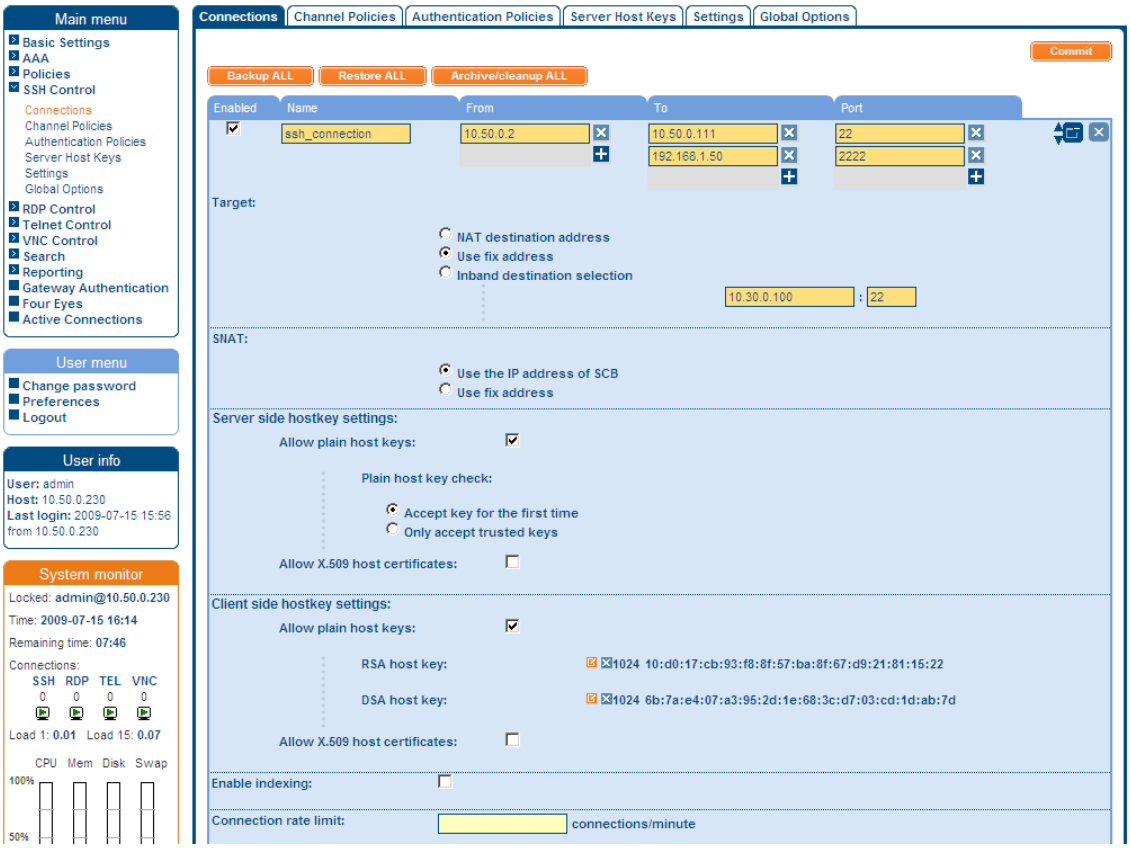


Figure 5.3. Configuring connections

Step 2. The **SNAT** section allows you to configure Source Network Address Translation (SNAT) on the server side of SCB. SNAT determines the IP address SCB uses in the server-side connection. The target server will see the connection coming from this address. The following options are available:

- **Use the IP address of SCB**: Server-side connections will originate from SCB's internal interface in Router and Bridge mode, and from the external interface in Bastion mode. This is the default behavior of the connection.
- **Use the original IP address of the client**: Server-side connections will originate from the client's IP address, as seen by SCB's external interface. This option not available in Bastion mode.
- **Use fix address**: Enter the IP address that will be used as the source address in server-side connections.

Warning
Do not forget to properly configure routers and other network devices when using the **Use fix address** option: messages sent by the server to this address must reach SCB.

Step 3. Click **Commit**.

5.1.4. Procedure – Creating and editing channel policies

Purpose:

The channel policy lists the channels (for example, terminal session and SCP in SSH; Drawing, Clipboard in RDP) that can be used in a connection. The channel policy can further restrict access to each channel based on the IP address of the client or the server, a user list, user group, or a time policy. For example, all clients may access the servers defined in a connection via SSH terminal, but the channel policy may restrict SCP access only to a single client. The policies set in the channel policy are checked when the user attempts to open a particular channel type in the connection.

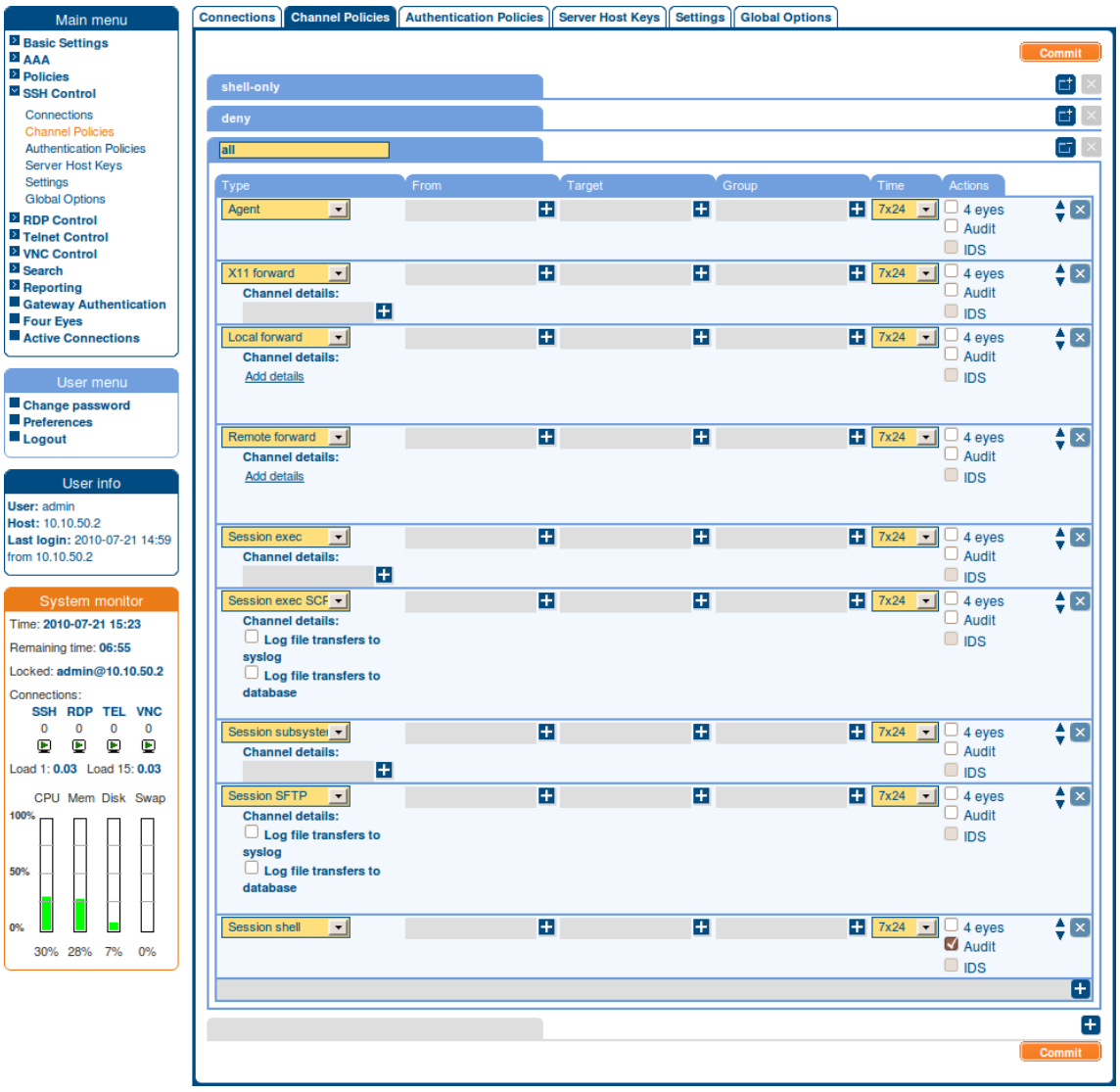


Figure 5.4. Configuring channel policies

To create a new channel policy or edit an existing one, complete the following procedure:

Steps:

Step 1. Channel policies are configured individually for every protocol. Navigate to the **Channel Policies** tab of the respective protocol (for example, **SSH Control > Channel Policies**) and click ✚ to create a new channel policy. Enter a name for the policy into the **Channel Policy** field (for example, `shell_and_backup`).

Step 2. Click ✚ to add a new channel.

Step 3. Select the channel to be enabled in the connection from the **Type** field. All restrictions set in the following steps will be effective on this channel type. The available channels are different for every protocol. For their descriptions, see the following sections:

- *Section 5.2.2, Supported SSH channel types (p. 151)* for the Secure Shell protocol

- *Section 5.3.1, Supported RDP channel types (p. 165)* for the Remote Desktop Protocol.
- The Telnet protocol has only one channel type with no special configuration options.
- The VNC protocol has only one channel type with no special configuration options.

Step 4. To restrict the availability of the channel only to certain clients, click ⊞ in the **From** field and enter the IP address of the client allowed to use this type of the channel. Repeat this step until all required client IPs are listed.

Step 5. To restrict the availability of the channel only to certain servers, click ⊞ in the **Target** field and enter the IP address of the server allowed to use this type of the channel. Repeat this step until all required server IPs are listed.

Note
Use the real IP address of the server, which may be different from the one addressed by the clients, specified in the **Target** field of the connection policy.

Step 6. To restrict the availability of the channel only to certain users, click ⊞ in the **Remote Group** field and enter the name of the user group allowed to use this type of the channel. Repeat this step until all permitted groups are listed.

You may list local user lists as defined in *Procedure 5.1.6, Creating and editing user lists (p. 130)*, or LDAP groups (for details on accessing LDAP servers from SCB, see *Procedure 5.1.7, Authenticating users to an LDAP server (p. 131)*). Note the following behavior of SCB:

- If you list multiple groups, members of any of the groups can access the channel.

Note
When listing both a whitelist and blacklist in the **Remote Group** section and a username appears on both lists, the user will be able to access the channel.

- If a local user list and an LDAP group has the same name and the LDAP server is configured in the connection that uses this channel policy, both the members of the LDAP group and the members of the local user list can access the channel.

Note
User lists and LDAP support is currently available only for the SSH and RDP protocols. For other protocols, see *Section 8.2, Configuring gateway authentication (p. 210)*.

Step 7. Select a time policy to narrow the availability of the channel. If the time policy of the channel policy is set to `7x24`, the channel is always available. For details, see *Procedure 5.1.5, Configuring time policies (p. 129)*.

Step 8. Some channel types require additional parameters, for example port forwarding in SSH needs the IP addresses and ports of the source and destination machines. Click ⊞ in the **Details** field and enter the required

parameters. For a list of parameters used by the different channels, see *Section 5.2.2, Supported SSH channel types (p. 151)* and *Section 5.3.1, Supported RDP channel types (p. 165)*.

Step 9. Select the **Audit** option to record the activities of the channel into audit trails. Typically large file-transfers (for example system backups, SFTP channels) are not audited because they result in very large audit trails. Check regularly the free hard disk space available on SCB if you do audit such channels. You can also receive alerts about disk space fill up if you set these. For details, see *Procedure 4.3.4.3, Preventing disk space fill up (p. 59)* and *Section 4.3.4.4, System related traps (p. 60)*.

Step 10. Select the **4-eyes** option to require 4-eyes authorization to access the channel. For details, see *Section 8.3, Configuring 4-eyes authorization (p. 215)*.

Step 11. Select the **IDS** option to forward the contents of the channel to an IDS or DLP system. For details, see *Procedure 5.1.11, Forwarding traffic to an IDS or DLP system (p. 144)*.

Step 12. Repeat Steps 2-11 to add other channels to the policy.

Step 13. Click **Commit** to save the list.

5.1.5. Procedure – Configuring time policies

Purpose:

The time policy determines the timeframe when the users are permitted to access a particular channel. By default, there is no time-based restriction, all channels are available 7x24. Complete the following procedure to create a time policy or edit an existing one:

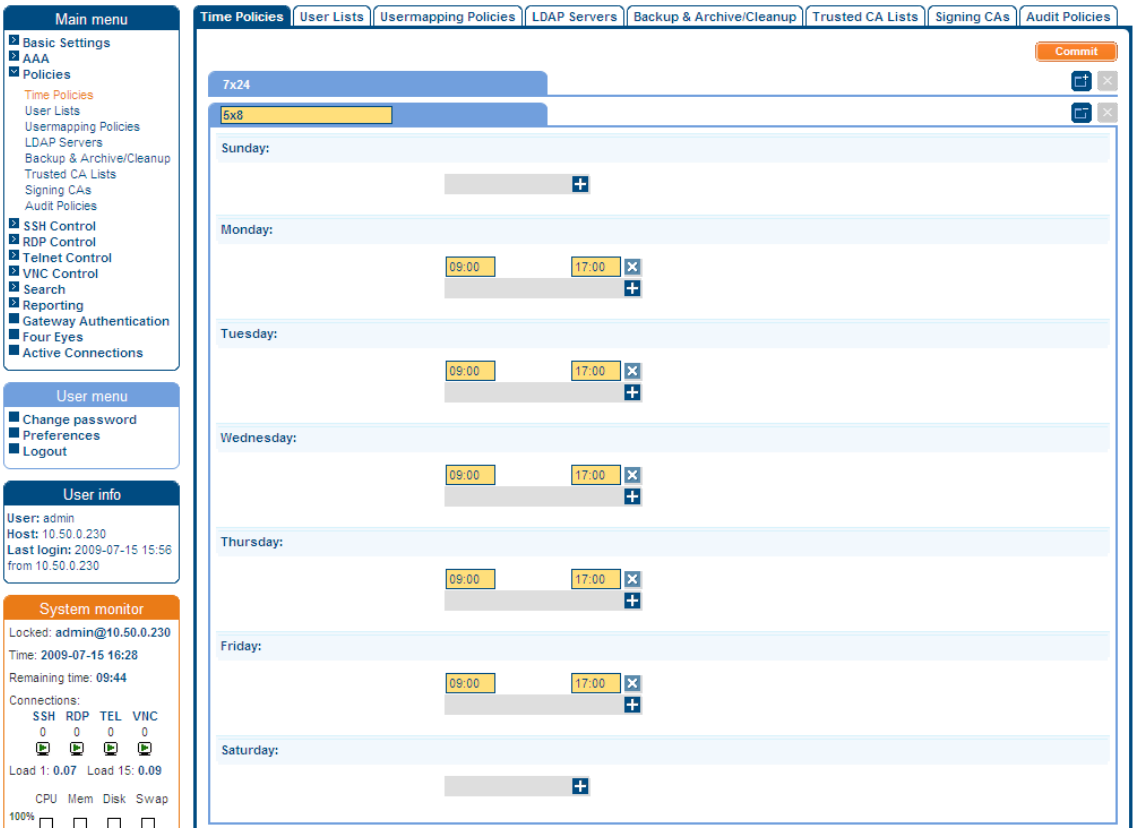


Figure 5.5. Configuring time policies

Steps:

Step 1. Navigate to the **Time Policies** tab of the **Policies** menu item and click ⊞ to create a new time policy. Enter a name for the policy into the **Time Policy** field (for example `workhoursonly`).

Step 2. Click ⊞ to display the days of the week and the allowed intervals.

Step 3. Enter the intervals for each day when the users are allowed to access the connection. Use the `hh:mm` format (for example from 08:00 to 16:00).

Step 4. To add multiple intervals for a day, click ⊞.

Step 5. Click **Commit**.

Step 6. To actually restrict access to a connection or a channel based on the policy created in the previous steps:

- Select this policy in the **Time Policy** field of the channel policy.
- Click **Commit**.

5.1.6. Procedure – Creating and editing user lists

Purpose:

User lists are white- or blacklists of usernames that allow fine-control over who can access a connection or a channel. Complete the following procedure to create a new user list or edit an existing one:

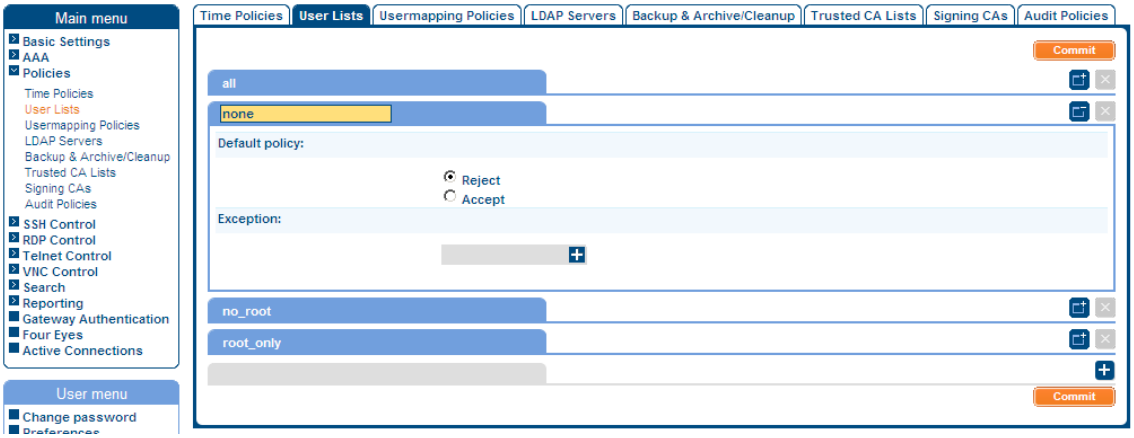


Figure 5.6. Configuring user lists

Steps:

Step 1. Navigate to the **User Lists** tab of the **Policies** menu and click ⊞ to create a new user list. Enter a name for the list into the **User List** field (for example `serveradmins`).

Step 2. Click ⊡ to display the list of users.

Step 3. Select the default policy of the user list. Select **Reject** for a whitelist, that is, to allow access only to the members of the list. Select **Accept** for a blacklist, that is, to allow access to everyone except the members of the list.

Step 4. Click ⊞ and enter a username into the displayed field. Repeat this step until all required usernames are listed.

Step 5. Click **Commit** to save the list.

Step 6. To actually restrict access to a channel based on the user list created in the previous steps:

- Navigate to the **Channel Policies** tab of the type of connection you want to control and click ⊡ to display the details of the policy.
- Click ⊞ in the **Group** section to add a new group to the policy and enter the name of the group. Repeat this step to add other groups.

Note
When listing more groups, users of any of the listed groups can access the channel. For details, see *Procedure 5.1.4, Creating and editing channel policies (p. 126)*.

When listing both a whitelist and blacklist in the **Group** section and a username appears on both lists, the user will be able to access the channel.

- Click **Commit**.

5.1.7. Procedure – Authenticating users to an LDAP server

Purpose:

Note
This feature is currently available only for SSH and RDP connections. For other protocols, see *Section 8.2, Configuring gateway authentication (p. 210)*.

SCB can authenticate the users of the controlled connections to LDAP databases. To authenticate the users to an LDAP database, complete the following steps.

Steps:

Step 1. Navigate to **Policies > LDAP Servers** and click ⊞ to create a new LDAP policy.

Step 2. Enter a name for the policy into the **LDAP Policy** field (for example `ldapservers`).

Step 3.

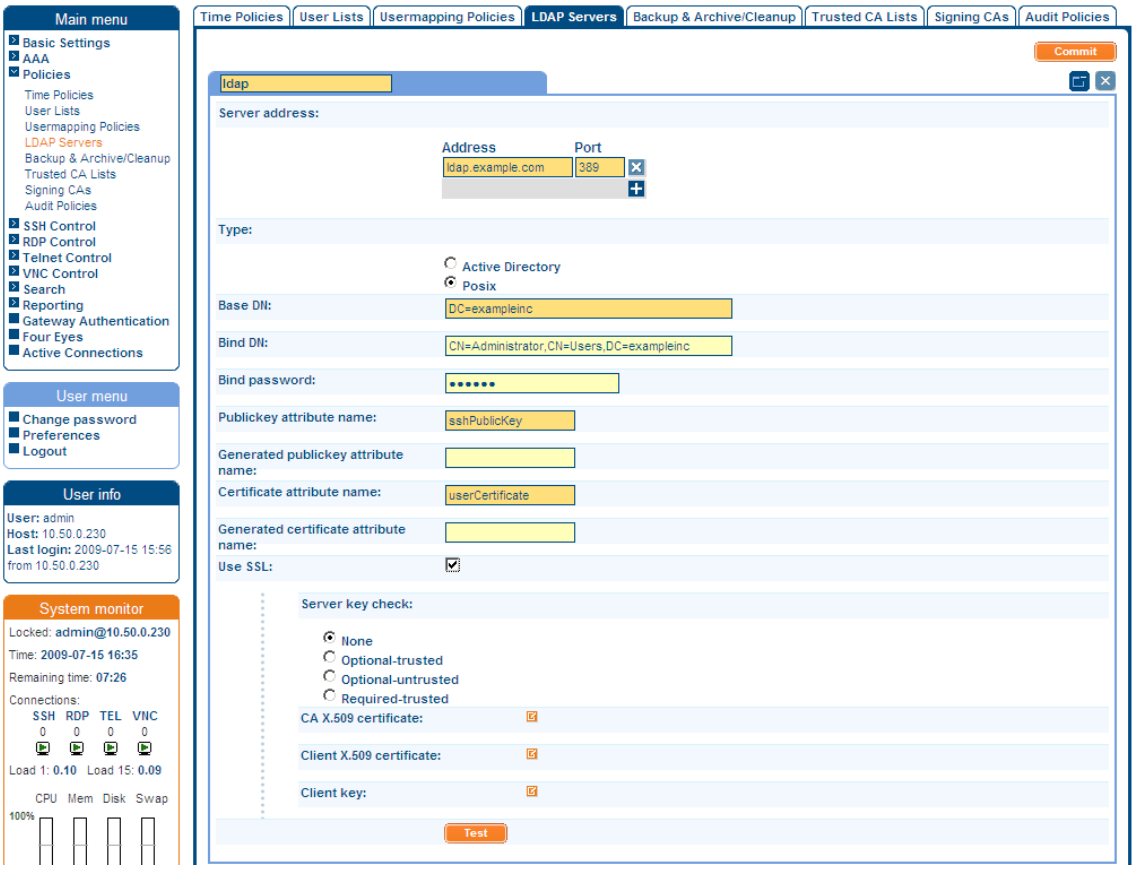


Figure 5.7. Configuring LDAP Server policies

Step a. Enter the IP address or hostname and port of the LDAP server into the **Server Address** field.

To add multiple servers, click ⊞ and enter the address of the next server. If a server is unreachable, SCB will try to connect to the next server in the list in failover fashion.

Warning
If you will use a TLS-encrypted with certificate verification to connect to the LDAP server, use the full domain name (for example *ldap.example.com*) in the **Server Address** field, otherwise the certificate verification might fail. The name of the LDAP server must appear in the Common Name of the certificate.

Step b. Select the type of your LDAP server in the **Type** field. Select **Active Directory** to connect to Microsoft Active Directory servers, or **Posix** to connect to servers that use the POSIX LDAP scheme.

Step c. Enter the name of the DN to be used as the base of the queries into the **Base DN** field (for example *DC=demodomain,DC=exampleinc*).

Step d. Enter the name of the DN where SCB should bind to before accessing the database into the **Bind DN** field.
For example: *CN=Administrator,CN=Users,DC=demodomain,DC=exampleinc*.

Note
SCB accepts both pre-win2000-style and Win2003-style account names (User Principal Names), for example *administrator@example.com* is also accepted.

Step e. Enter the password to use when binding to the LDAP server into the **Bind Password** field.

Step 4. *Skip this step if you use passwords to authenticate the users.*

- If you use public-key authentication and receive the public key of the users from the LDAP database, enter the name of the LDAP attribute that stores the public keys of the users into the **Publickey attribute name** field. For details on using public-key authentication with the LDAP database, see *Section 9.1, Configuring public-key authentication on SCB (p. 220)*.
- If you use X.509 certificate for authentication and receive the certificates of the users from the LDAP database, enter the name of the LDAP attribute that stores the certificates of the users into the **Certificate attribute name** field.

Step 5. *Skip this step if you use passwords to authenticate the users.*

- If you use public-key authentication and want SCB to generate server-side encryption keys on-the-fly and store them in a separate attribute on the LDAP server, enter the name of the attribute into the **Generated publickey attribute name** field.
- If you use certificate authentication and want SCB to generate server-side certificates on-the-fly and store them in a separate attribute on the LDAP server, enter the name of the attribute into the **Generated certificate attribute name** field.

Step 6. If you want to encrypt the communication between SCB and the LDAP server, select the **Use StartTLS** option and complete the following steps:

Note
TLS-encrypted connection to Microsoft Active Directory is supported only on Windows 2003 Server and newer platforms. Windows 2000 Server is not supported.

Only the STARTTLS method is supported to encrypt the communication between the LDAP server and SCB. The LDAPS protocol is not supported.

- If you want SCB to verify the certificate of the server, click the icon in the **CA X.509 certificate** field. A popup window is displayed.

  Click **Browse**, select the certificate of the Certificate Authority (CA) that issued the certificate of the LDAP server, then click **Upload**. Alternatively, you can paste the certificate into the **Copy-paste** field and click **Set**.

  SCB will use this CA certificate to verify the certificate of the server, and reject the connections if the verification fails.

  If you want SCB to simply accept any certificate shown by the LDAP server, select **Required untrusted** in the **Server key check** field.

Note
Alternatively, you can use the following options to check the certificate of the LDAP server:

- *None*: Do not request a certificate from the remote host, and accept any certificate if the host sends one.
- *Optional trusted*: If the remote host sends a certificate, SCB checks if it is valid (not expired) and that the Common Name of the certificate contains the domain name or the IP address of the host. If these checks fail, SCB rejects the connection. However, SCB accepts the connection if the host does not send a certificate.
- *Optional untrusted*:Accept any certificate shown by the remote host. Note that the host must show a certificate.
- *Required trusted*: Verify the certificate of the remote host. Only valid certificates signed by a trusted certificate authority are accepted. For details on importing CA certificates, see *Procedure 4.5.7.2, Uploading external certificates to SCB (p. 118)*. Note that the Common Name of the certificate must contain the domain name or the IP address of the host.
- *Request untrusted*: SCB requests a certificate from the remote host, and rejects the connection if no certificate is received; if the certificate is not valid (expired); or if the Common Name of the certificate does not contain the domain name or the IP address of the host.

Warning
If you will use a TLS-encrypted with certificate verification to connect to the LDAP server, use the full domain name (for example `ldap.example.com`) in the **Server Address** field, otherwise the certificate verification might fail. The name of the LDAP server must appear in the Common Name of the certificate.

- If the LDAP server requires mutual authentication, that is, it expects a certificate from SCB, generate and sign a certificate for SCB, then click in the **Client X.509 certificate** field to upload the certificate. After that, click in the **Client key** field and upload the private key corresponding to the certificate.

Step 7. If you have modified the **Server key check** field or the keys used in the connections, perform the following steps:

Warning
This step terminates all controlled connections going through SCB. Disconnect your clients from the protected servers before proceeding.

To activate the new settings, navigate to **Basic Settings > System > Traffic control** section and click **Restart**.

5.1.8. Audit policies

An audit trail is a file storing the recorded activities of the administrators. Audit trails can be replayed using the Audit Player application (for details, see *Chapter 7, Viewing session information and replaying audit trails (p. 191)*). The audit trails are automatically compressed, and can be optionally encrypted, timestamped, and signed as well. Note that audit trails are not created automatically for every connection, auditing must be enabled manually in the channel policy used in the connection. However, the default channel policies available enable auditing for the most common channels.

Tip
By default, every connection uses the built-in **default** audit policy. Unless you use a custom audit policy, modifying the **default** audit policy will affect every audited channel of the connections passing SCB.

Note
In SCB 1.x it was possible to compress the audit trail files, and also to set the level of compression. Starting from version 2.0, the audit trails are automatically compressed.

- For details on how to configure audit trail encryption, see *Procedure 5.1.8.1, Encrypting audit trails (p. 135)*.
- For details on how to configure timestamping, see *Section 5.1.8.2, Timestamping audit trails (p. 138)*.
- For details on how to configure audit trail signing, see *Procedure 5.1.8.3, Digitally signing audit trails (p. 140)*.
- For details on how to configure other audit trail options, see *Procedure 5.1.8.4, Limiting audit trails (p. 141)*.

5.1.8.1. Procedure – Encrypting audit trails

Purpose:

SCB can encrypt the audit trails to prevent unauthorized access to the audit trail files. Encryption requires one or more X.509 certificates. Note that SCB itself cannot create the certificates used to encrypt the audit trails.

SCB has the following different ways to encrypt the audit trails:

- *Encrypt the audit trails with a single certificate.* This is the most simple approach: SCB uses one certificate to encrypt the audit trails; anyone who has the private key of that certificate can replay the audit trails. If that key is lost, there is no way to open the audit trails.
- *Encrypt the audit trails separately with multiple certificates.* SCB uses two or more certificates separately to encrypt the audit trails; anyone who has the private key of one of the encryption certificates can replay the audit trails.
- *Encrypt the audit trails jointly with two certificates.* SCB uses two certificates together (a certificate-pair) to encrypt the audit trails. The private keys of both encryption certificates are needed to replay the audit trails. This is a kind of "4-eyes in auditing".
- *Use a separate certificate to encrypt the upstream traffic.* SCB can use a different certificate to encrypt the upstream traffic that contains sensitive information, for example the passwords typed by the user in a connection are visible in the upstream traffic. The upstream traffic will be displayed only if the private key of this certificate is available.

Note
Even if the upstream traffic is encrypted with a separate certificate, only one audit trail file is created for a session.

Warning
Sensitive information like passwords of the user logging in to the remote server are visible in the audit trails, unless the **Use a separate certificate to encrypt the upstream traffic** option is used. In this case, the passwords are visible only with the private key of the certificate used for encrypting the upstream traffic.

Note
You can combine the different encryption methods, so for example it is possible to encrypt the audit trails with multiple certificate-pairs, and to replay the trails only if the private keys of a certificate-pair are available. This is true for encrypting the upstream traffic as well. At the extreme, you will need four private keys to fully replay an audit trail: two to open the normal traffic, and two more to display the upstream traffic.

For details on uploading certificates to SCB, see *Procedure 4.5.7.2, Uploading external certificates to SCB (p. 118).*

Tip
If two certificates are displayed in a row, they are a certificate-pair and you need the private key of both to replay the audit trails. If two certificates are displayed in separate rows, you need the one of the private keys to replay the audit trails. If there are multiple rows containing two certificates, you need the private keys of the certificate(s) listed in any of the rows.

Steps:

Step 1.

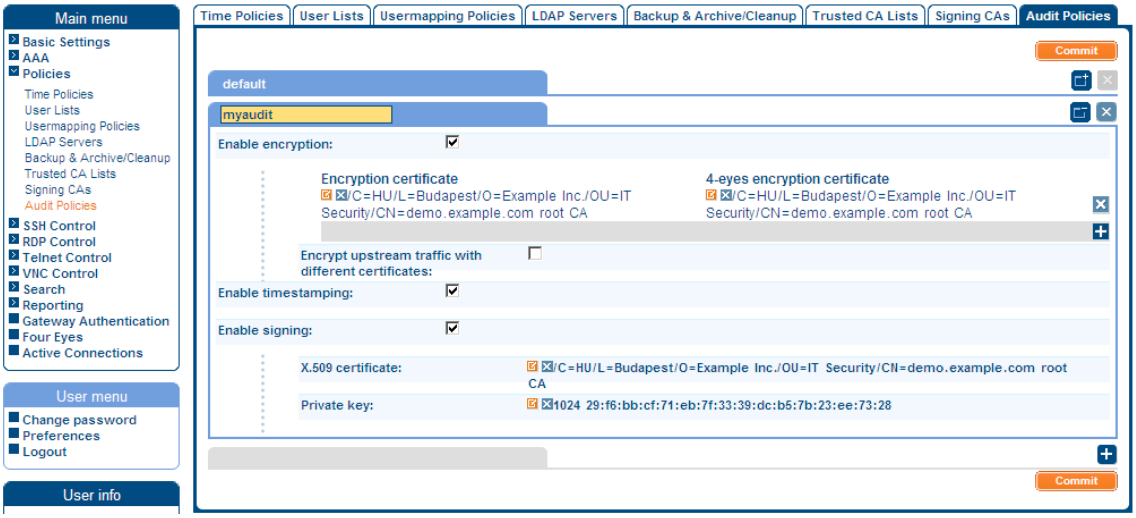


Figure 5.8. Encrypting audit trails

Navigate to **Policies > Audit Policies** and select the audit policy you will use in your connections.

Tip
The **default** policy is used by default by all added connections. Change this policy if you want to encrypt every audited connection.

Step 2. Select the **Enable encryption** option.

Step 3. Click ⊞ in the **Encryption X.509 certificates** field. A new row is displayed.

Step 4. Click on the left ⊠ icon and upload a certificate to SCB. This certificate will be used to encrypt the audit trails, and it must not include the private key.

Note
To replay the audit trails, you will need the private key of the certificate on the computer running the Audit Player application.

Step 5. *Optional step*: To encrypt the audit trails jointly with another certificate, click on the right ⊠ icon and upload a certificate to SCB. Note that the private key of both certificates will be required to replay the audit trails.

Step 6. *Optional step*: Repeat Steps 3-4 (and optionally Step 5) to encrypt the audit trails separately with additional certificates.

Step 7. *Optional step*: To use a separate certificate to encrypt the upstream traffic, complete the following steps:

Step a. Select **Different encryption certificates for upstream**.

Step b. Click the left ⊞ icon in the **Upstream encryption X.509 certificates** field and upload a certificate.

Step c. *Optional steps*: If needed, add further certificate to jointly or separately encrypt the upstream traffic with.

Step 8. Click **Commit**.

5.1.8.2. Timestamping audit trails

SCB can add timestamps to the audit trails.

- For details on how to use the built-in timestamping service of SCB, see *Procedure 5.1.8.2.1, Built-in timestamping service (p. 138)*.
- For details on how to use an external Timestamping Authority, see *Procedure 5.1.8.2.2, External timestamping service (p. 139)*.

5.1.8.2.1. Procedure – Built-in timestamping service

Purpose:

To use the built-in timestamping service of SCB, complete the following steps:

Steps:

Step 1. Navigate to **Policies > Audit Policies** and select the audit policy you will use in your connections.

Tip
The **default** policy is used by default by all added connections. Change this policy if you want to encrypt every audited connection.

Step 2.

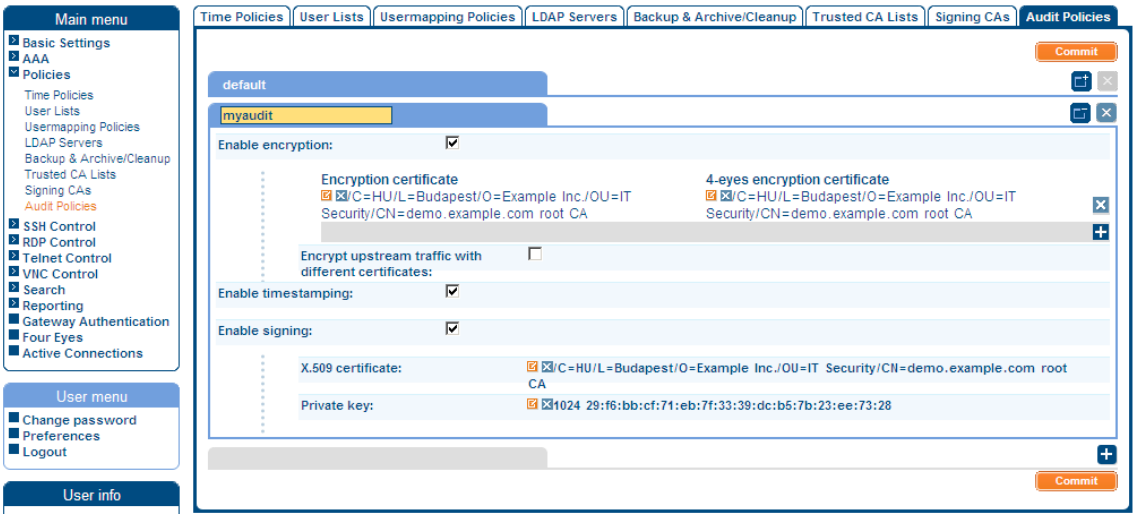


Figure 5.9. Timestamping audit trails

Select the **Enable timestamping** option.

Step 3. Click **Commit**. SCB will automatically add timestamps to the audit trails of every connection that is audited and uses this audit policy.

Note
For details on how to change the certificate used for timestamping, see *Section 4.5.7, Managing the certificates used on SCB (p. 116).*

Step 4. Repeat the above steps for other audit policies if needed.

5.1.8.2.2. Procedure – External timestamping service

Purpose:

To request timestamps from a remote Timestamping Authority (TSA) , complete the following steps:

Steps:

Step 1.

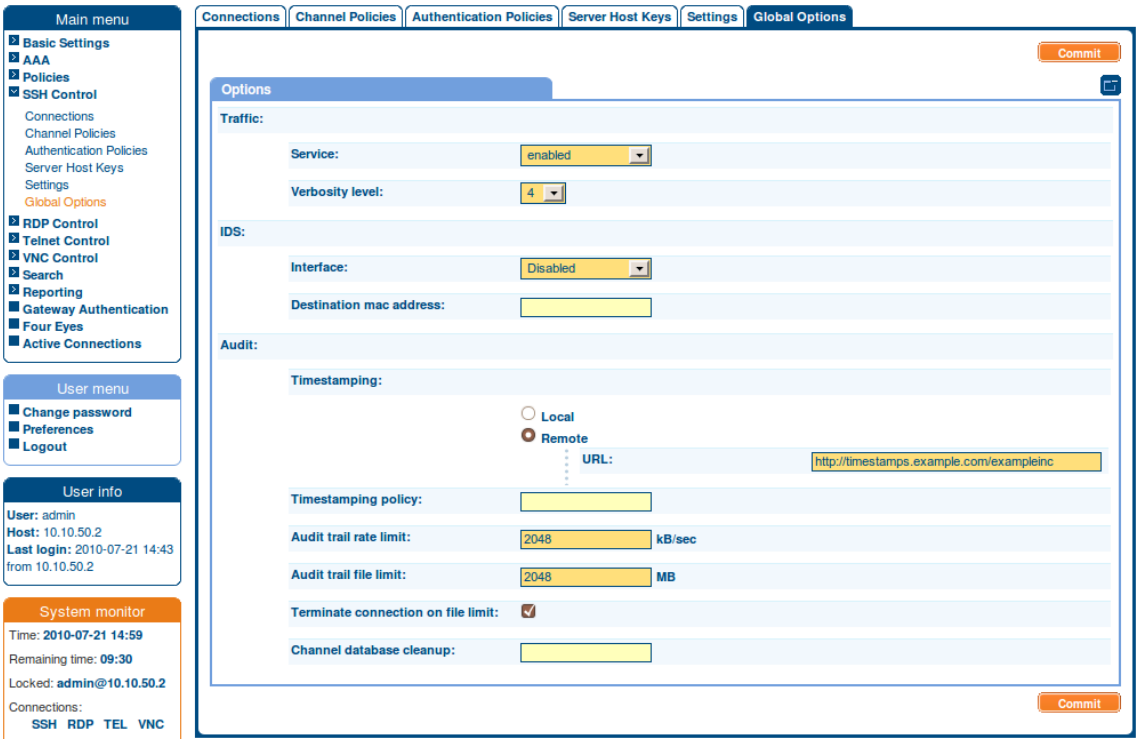


Figure 5.10. Configuring a remote TSA

Navigate to **SSH Control > Global Options > Timestamping** and select **Remote**.

Step 2. Enter the address of the timestamping server into the **URL** field. Note that currently only plain HTTP services are supported, password-protected and HTTPS services are not supported at.

Step 3. If the Timestamping Server has timestamping policies configured, enter the OID of the policy to use into the **Timestamping policy** field. SCB will include this ID in the timestamping requests sent to the TSA.

Step 4. Click **Commit**.

Step 5. Repeat the above steps for other protocols if needed.

Step 6. Navigate to **Policies > Audit Policies** and select the audit policy you will use in your connections.

Tip
The **default** policy is used by default by all added connections. Change this policy if you want to encrypt every audited connection.

Step 7. Select the **Timestamping** option.

Step 8. Click **Commit**. SCB will automatically add timestamps to the audit trails of every connection that is audited and uses this audit policy.

Step 9. Repeat the above steps for other audit policies if needed.

5.1.8.3. Procedure – Digitally signing audit trails

Purpose:

SCB can digitally sign the audit trails to prevent the manipulation of the audit trail files. This requires an X.509 certificate and also the private key of the certificate. Note that SCB can generate a private key that you can use to create a certificate, but SCB itself cannot create the certificate used to sign the audit trails.

Steps:

Step 1.

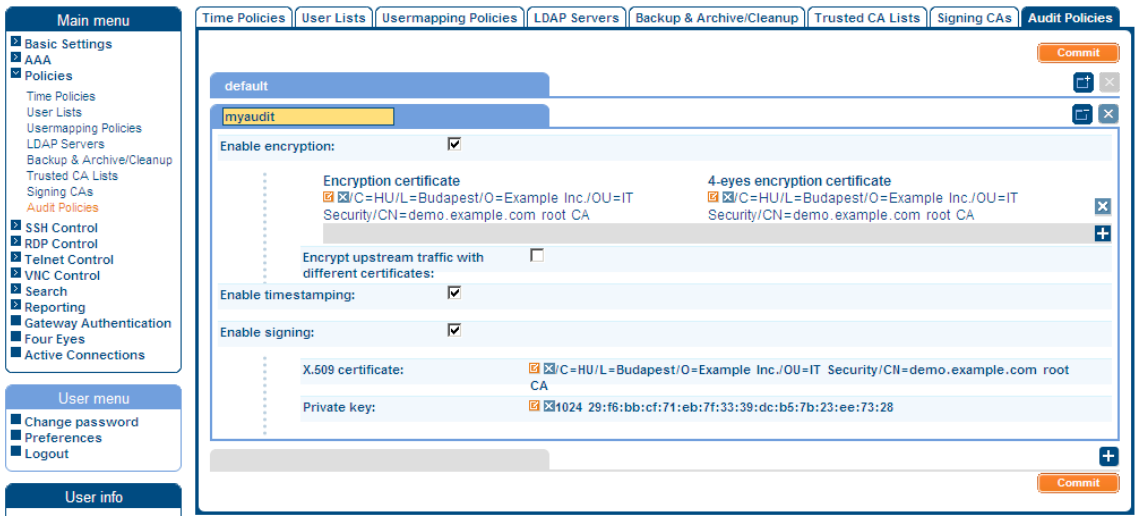


Figure 5.11. Signing audit trails

Navigate to **Policies > Audit Policies** and select the audit policy you will use in your connections.

Tip
The **default** policy is used by default by all added connections. Change this policy if you want to encrypt every audited connection.

Step 2. Select the **Signing** option.

Step 3. Upload a certificate and the corresponding private key to SCB. For details, see *Procedure 4.5.7.2, Uploading external certificates to SCB (p. 118)*.
To generate a key-pair on SCB and use it to create a certificate in your PKI system, complete the following steps:

Step a. Click the icon in the **Private key** field. A popup window opens.

Step b. Select the length of the key in the **Generate key** field.

Step c. Set the type of the key (RSA or DSA) in the **Key type** field.

Step d. Click **Generate**. The fingerprint of the new private key is displayed in the **Private key** field.

Step e. Click **Commit**.

Step f. Click the fingerprint and select a format to download the public part of the key. Currently the key can be downloaded in the following formats: `OpenSSH`, `Tectia`, `PEM`, `DER`.

Step g. Use your PKI system to create a certificate request with the downloaded key, and sign it with your Certificate Authority (CA).

Step h. Click in the **X.509 certificate** field and upload the certificate to SCB.

Step 4. Click **Commit**. SCB will automatically sign the audit trails of every connection that is audited and uses this audit policy.

Step 5. Repeat the above steps for other audit policies if needed.

5.1.8.4. Procedure – Limiting audit trails

Purpose:

To impose limits on the audit trail files, complete the following steps:

The followings must be set for every controlled protocol separately.

Steps:

Step 1. Navigate to **Control > Global options > Audit**.

Step 2. Adjust the **Audit trail rate limit** option if needed. If the size of an audit trail is increasing faster than this rate, SCB can send an alert to the administrator. Such connections may sign Denial of Service (DOS) attacks. Different protocols can have different rate limits.

Step 3. Adjust the **Audit trail file limit** option if needed. The size of audit trail files is maximized to prevent a single connection from filling SCB's hard drive, causing a Denial of Service (DOS). The maximal size of the **Audit trail file limit** can be 4GB.

Step 4. Enable the **Terminate connection on file limit** option to terminate any connection if the size of its audit trail reaches limit specified in **Audit trail file limit**. If the **Terminate connection on file limit** option is disabled, the connection will not be terminated, but SCB will not further audit the connection. Different protocols can have different size limits.

Step 5. Click **Commit**.

Step 6.

Warning
This step terminates all controlled connections going through SCB. Disconnect your clients from the protected servers before proceeding.

To activate the new settings, select **Basic Settings > System > Traffic control > Restart**.

5.1.9. Procedure – Verifying certificates with Certificate Authorities

Purpose:

SCB can check the validity of certificates using the certificates and certificate-revocation lists of the certificate authorities that issued the certificates. This can be used for example to verify the certificates of the servers in SSH/RDP connections. To create a list of CA certificates to use during the certificate validation, complete the following steps:

Steps:

Step 1.

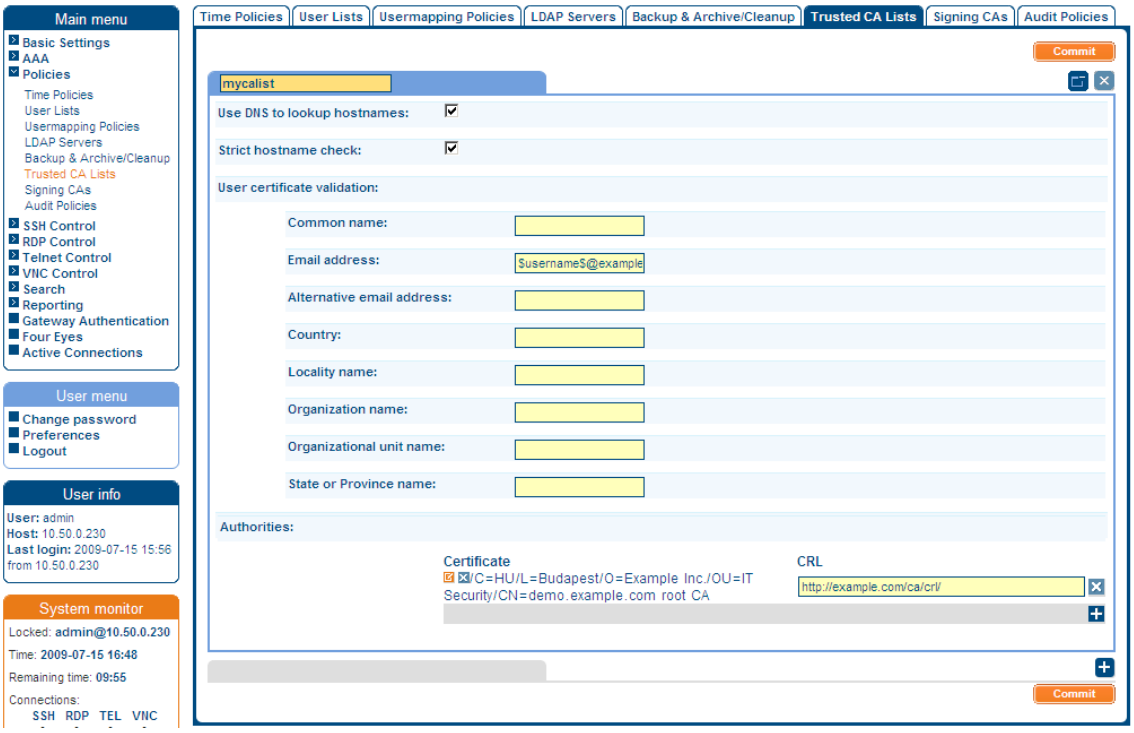


Figure 5.12. Creating Trusted CA lists

Navigate to **Policies > Trusted CA Lists** and click ⊞ to create a new list.

Step 2. Enter a name for the CA list into the topmost field.

Step 3. Click in the **Certificate** field, and upload the certificate of the Certificate Authority (CA) that will be used to validate the certificates. For details on uploading certificates, see *Procedure 4.5.7.2, Uploading external certificates to SCB (p. 118)*.

Step 4. Enter the URL of the Certificate Revocation List of the CA into the **CRL** field. Certificates appearing on the CRL list will be automatically rejected.

Note
Note that only `.pem` format CRLs are accepted. CRLs that are in PKCS7 format (`.crl`) are not accepted.

Step 5. To further limit which certificates are accepted, you may use the following options:

- *Strict hostname check*: Select this option to accept only certificates when the Common Name of the certificate contains the hostname or the IP address of the host showing the certificate.
- *Use DNS to lookup hostnames*: Select this option to use the domain name server set on **Basic Settings > Network > Naming** to resolve the hostnames and IP addresses for certificate validation. If you have enabled the **Strict hostname check** option, you probably want to enable this option as well.
- To restrict the accepted certificates based on the content of the certificate, enter the required value into the appropriate field of the **User certificate validation** section. For example, to accept only certificates that contain `Example Inc.` in their Organization Name field, enter `Example Inc.` in to the **Organization Name** field. In the Common name, E-mail address, and Alternative e-mail address fields you can use the `$username` macro to refer to the username used in the connection. This macro makes it possible to check that the user is using his own certificate.

Step 6. Click **Commit**.

5.1.10. Procedure – Signing certificates on-the-fly

Purpose:

At a number of places, SCB can generate the server-side certificates on the fly. This technique is used for example in SSL-encrypted RDP sessions, RDP sessions that use Network Layer Authentication (CredSSP), or SSH connections that use X.509-based authentication. To create a signing CA, complete the following steps:

Note
Signing CAs require a CA certificate permitted to sign certificates, and also the corresponding private key.

Note
These CAs cannot be used to sign audit trails. For details on how to configure the certificates used to sign audit trails, see *Procedure 5.1.8.3, Digitally signing audit trails (p. 140)*.

Steps:

Step 1.

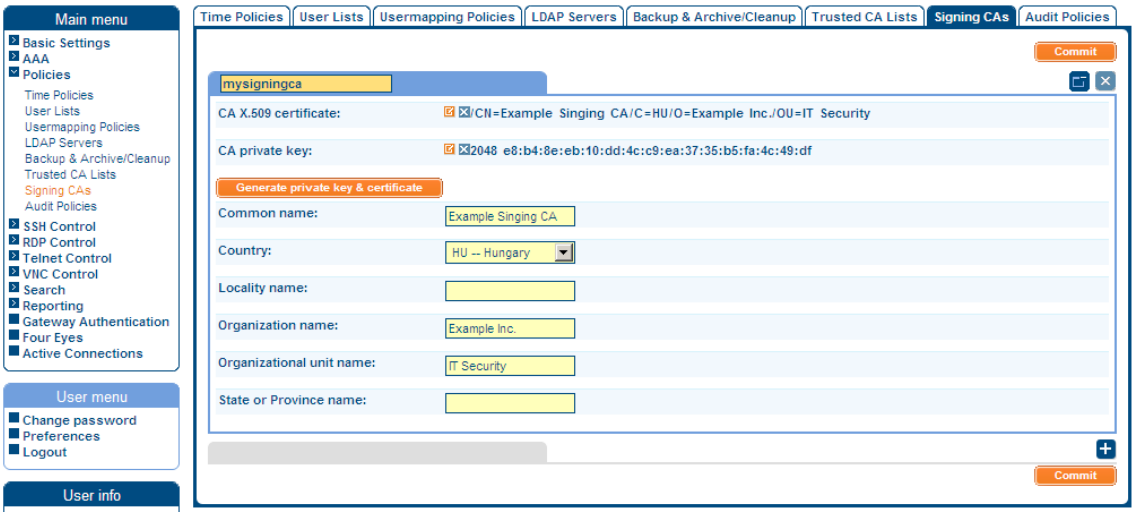


Figure 5.13. Creating Signing CAs

Navigate to **Policies > Signing CAs** and click ⊞.

Step 2. Enter a name for the CA into the topmost field.

Step 3. To upload a CA certificate and its private key, complete the following steps. Skip this step if you want to generate a CA on SCB.

Step a. Click ✎ in the **CA X.509 certificate** field and upload the certificate of the certificate authority.

Step b. Click ✎ in the **CA private key** field and upload the private key of the certificate authority. For details, see *Procedure 4.5.7.2, Uploading external certificates to SCB (p. 118)*.

Step c. Click **Commit**.

Step 4. To generate a CA certificate on SCB, complete the following steps:

Step a. Enter the Common Name for the CA certificate into the **Common Name** field. This name will be visible in the `Issued By` field of the certificates signed by this CA.

Step b. Fill the other fields as required, then click **Generate private key and certificate**.

Step c. Click **Commit**.

5.1.11. Procedure – Forwarding traffic to an IDS or DLP system

Purpose:

SCB can forward the contents of the traffic to an Intrusion Detection System (IDS) or a Data Leak Prevention (DLP) system for further analysis. The IDS or DLP system and SCB must be on the same network segment (connected to the same network switch). SCB does not modify the traffic, only forwards the unencrypted traffic to the IDS/DLP. For example, if HTTP traffic is tunneled in SSH, the IDS will receive only the HTTP traffic. That way, the IDS/DLP system can inspect the contents of the encrypted traffic. To configure traffic forwarding, complete the following steps.

Steps:

Step 1.

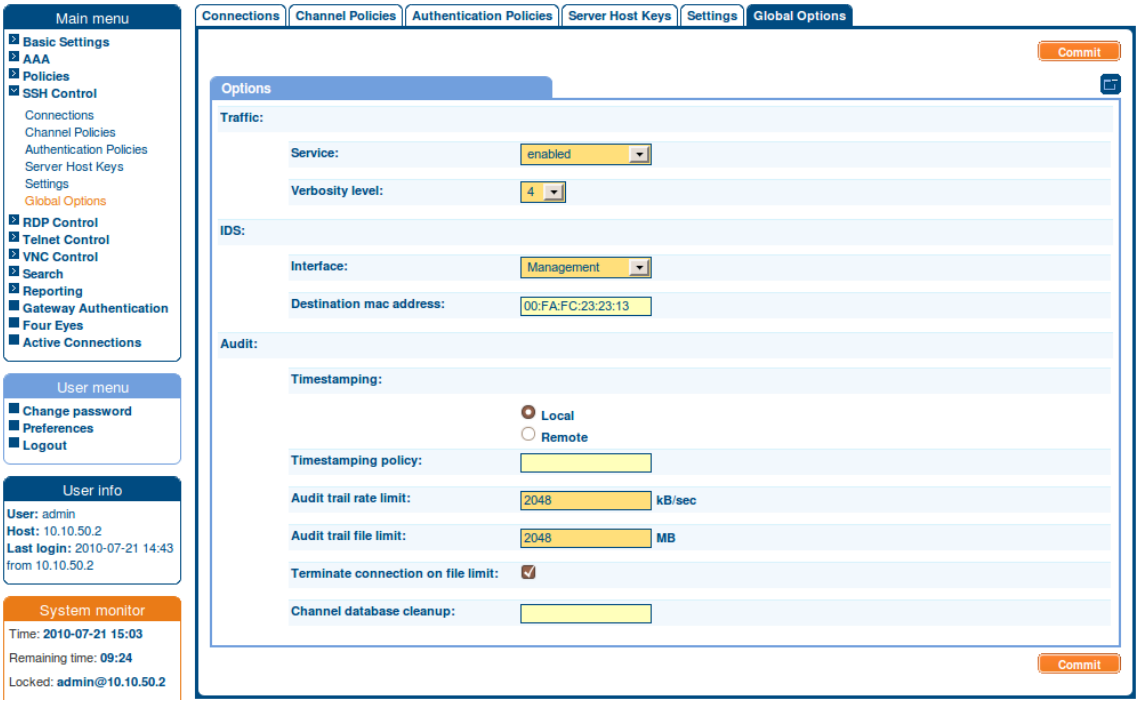


Figure 5.14. Configuring IDS-forwarding

Navigate to the type of traffic to forward (for example **SSH Control**), and select the **Global Options** page.

Step 2. Select **IDS > Interface**, and select which interface of SCB is connected to the same network segment (that is, to the same switch) as the IDS. Note that the HA interface of SCB cannot be used for this purpose.

Step 3. Enter the MAC address of the network card of the IDS system into the **Destination MAC address** field.

Step 4. Click **Commit**.

Step 5.

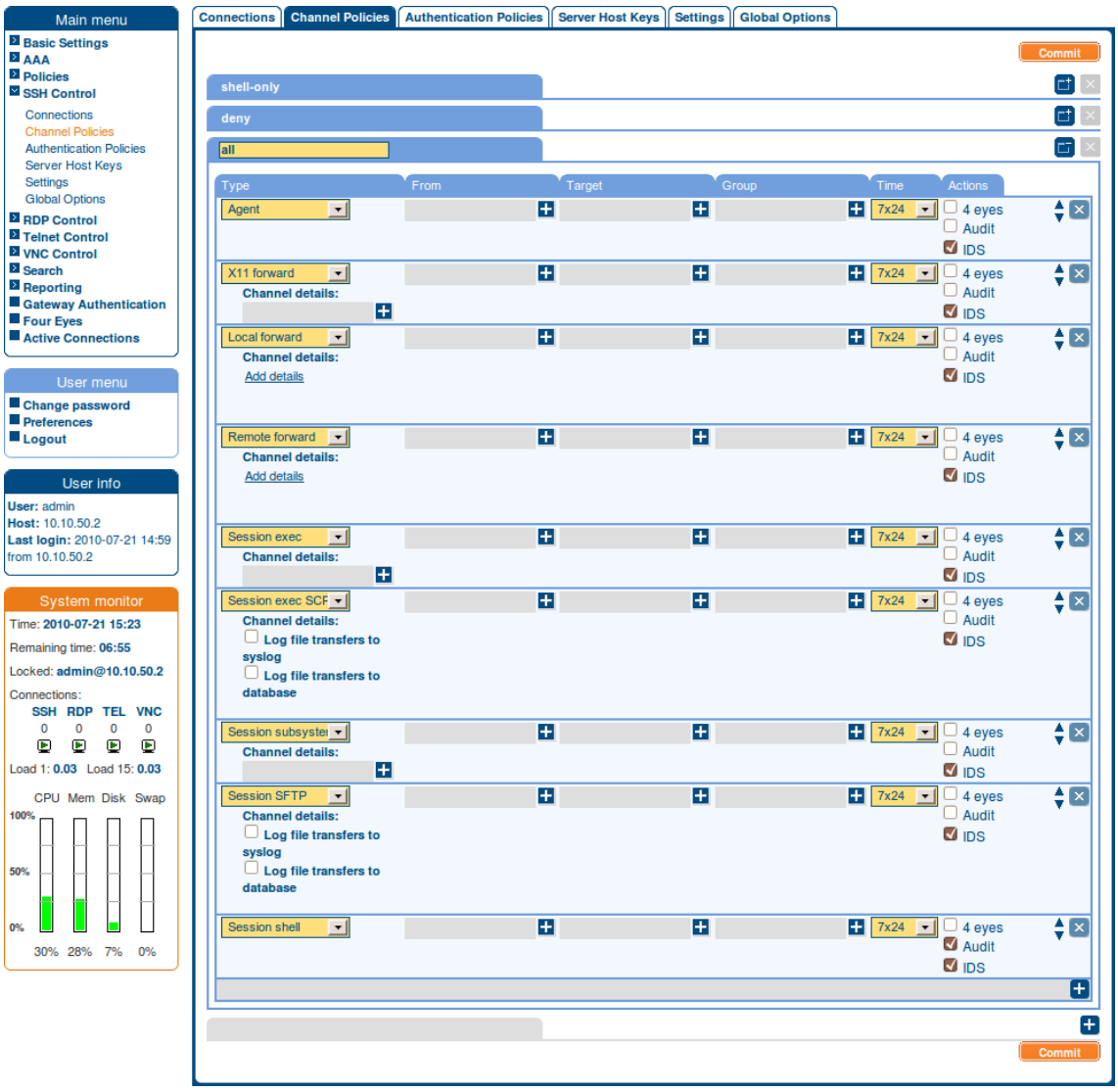


Figure 5.15. Enabling IDS-forwarding

Select **Channel Policies** page, and select the channel policy you use in the connections that you want to forward to the IDS. Click .

Step 6. Select the **IDS** option for the channels you want to forward to the IDS.

Step 7. Repeat the Steps 5-6 for other channel policies if needed.

Step 8.

Warning
This step terminates all controlled connections going through SCB. Disconnect your clients from the protected servers before proceeding.

Navigate to **Basic Settings > System > Traffic Control**, and click **Restart** for this type of traffic.

Step 9. Repeat the above steps for other types of traffic if needed.

5.1.12. Procedure – Configuring cleanup for the SCB connection database

Purpose:

SCB can automatically archive audit trails older than a specified retention time. However, the metadata of the corresponding connections is not deleted from the SCB connection database. Deleting the stored data about old connections decreases the size of the database, making searches faster, and might be also required by certain policies or regulations. The period after metadata is deleted can be specified individually for the different protocols, (for example, data about SSH connections can be stored longer than other connections) and also for every connection policy. In order to configure SCB to delete the metadata of old connections for a particular protocol, complete the following steps:

Steps:

Step 1. Navigate to the **Global Options** page of the respective protocol, for example, to *SSH Control > Global Options*.

Step 2.

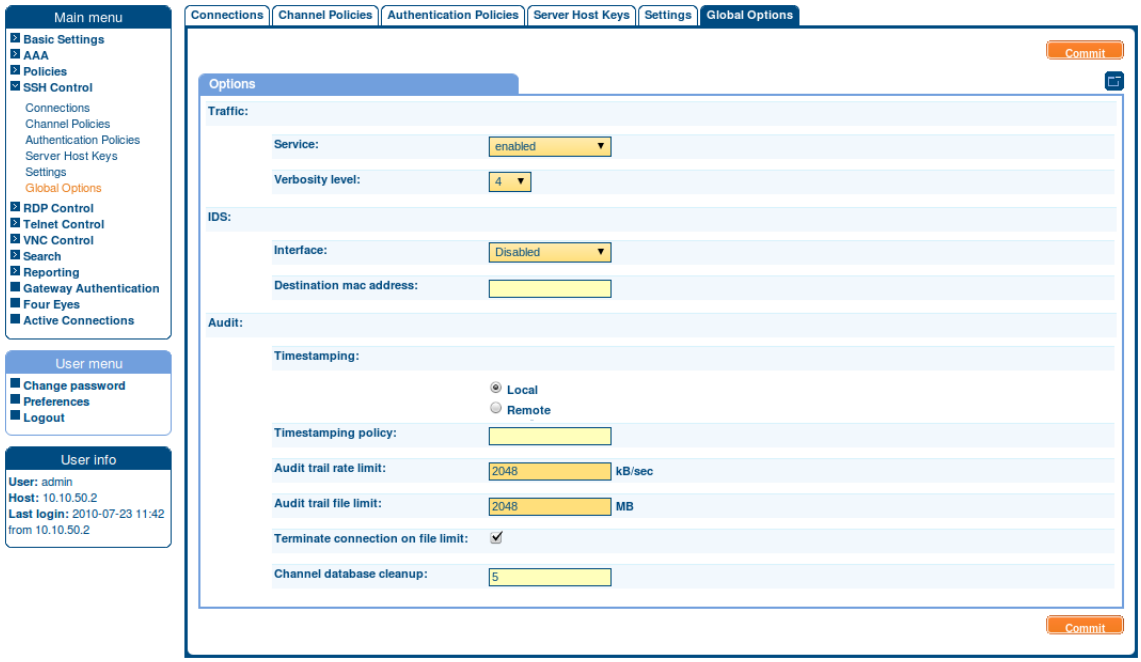


Figure 5.16. Configuring connection database cleanup for a protocol

Enter how long SCB (in days) should keep the metadata into the **Channel database cleanup** field. For example, if you specify *365*, SCB will delete the data of connections older than a year. Enter zero (*0*) to keep the data indefinitely (this is also the default behavior of SCB).

Note
The time you specify cannot be shorter than the **Retention time in days** set for the Archive policies used in the connections of this protocol.

The time you specify cannot be shorter than the **Channel database cleanup** set in the individual connection policies of this protocol.

Step 3. Click **Commit** and repeat the previous step for other protocols if needed.

Step 4.

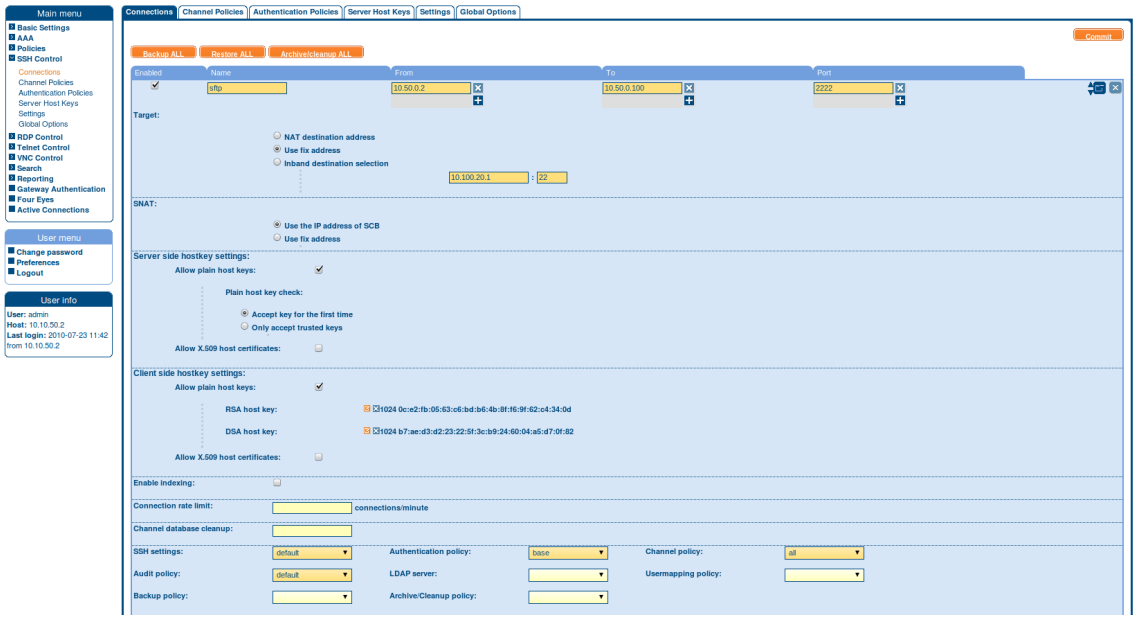


Figure 5.17. Configuring connection database cleanup for a connection

To delete the metadata of certain connections earlier than the time set in the **Global Options > Channel database cleanup** field of the protocol, navigate to the particular connection policy, and enter how long SCB (in days) should keep the metadata of the sessions of this connection policy into the **Channel database cleanup** field. Enter zero (*0*) to use the settings of the protocol (this is also the default behavior of SCB).

Note
The time you specify cannot be shorter than the **Retention time in days** set for the Archive policy used in the particular connection.

Step 5. Click **Commit** and repeat the previous step for other connections if needed.

Expected outcome:

Every day SCB deletes the metadata of connections older than the given cleanup time from the connection database.

5.2. SSH-specific settings

The following sections describe configuration settings available only for the SSH protocol. Use the following policies to control who, when, and how can access the SSH connection.

- *Hostkeys and host certificates*: SCB allows you to set how the identity of the client hosts and servers is verified. For details, see *Procedure 5.2.1, Setting the SSH host keys and certificates of the connection (p. 149)*.
- *Authentication Policy*: Authentication policies describe the authentication methods allowed in a connection. Different methods can be used for the client and server-side connections. For details, see *Section 5.2.3, Authentication Policies (p. 154)*.

- *User List*: A user list is a list of usernames permitted to use — or forbidden from using — the connection. Essentially it is a blacklist or a whitelist. All users matching the other requirements of the connection are accepted by default. For details, see *Procedure 5.1.6, Creating and editing user lists (p. 130)*.

- *Channel Policy*: The channel policy determines which SSH channels (for example terminal session, SCP, and so on) can be used in the connection, and whether they are audited or not. The different channels may be available only under certain restrictions, as set in the channel policy. For details, see *Procedure 5.1.4, Creating and editing channel policies (p. 126)*.

- *SSH settings*: SSH settings determine the parameters of the connection on the protocol level, including timeout value and greeting message of the connection. The following parameters determine which algorithms are used in the connections, and can be set independently for the client and the server side: key exchange, host key, cipher, MAC, and compression algorithms. The default values include all possible algorithms. For details, see *Procedure 5.2.5, Creating and editing protocol-level SSH settings (p. 163)*.

5.2.1. Procedure – Setting the SSH host keys and certificates of the connection

Purpose:

By default, SCB accepts and stores the host key or certificate of the server when the connection is first established. To manually set the SSH keys and certificates used and accepted in the connection, complete the following steps.

Steps:

Step 1.

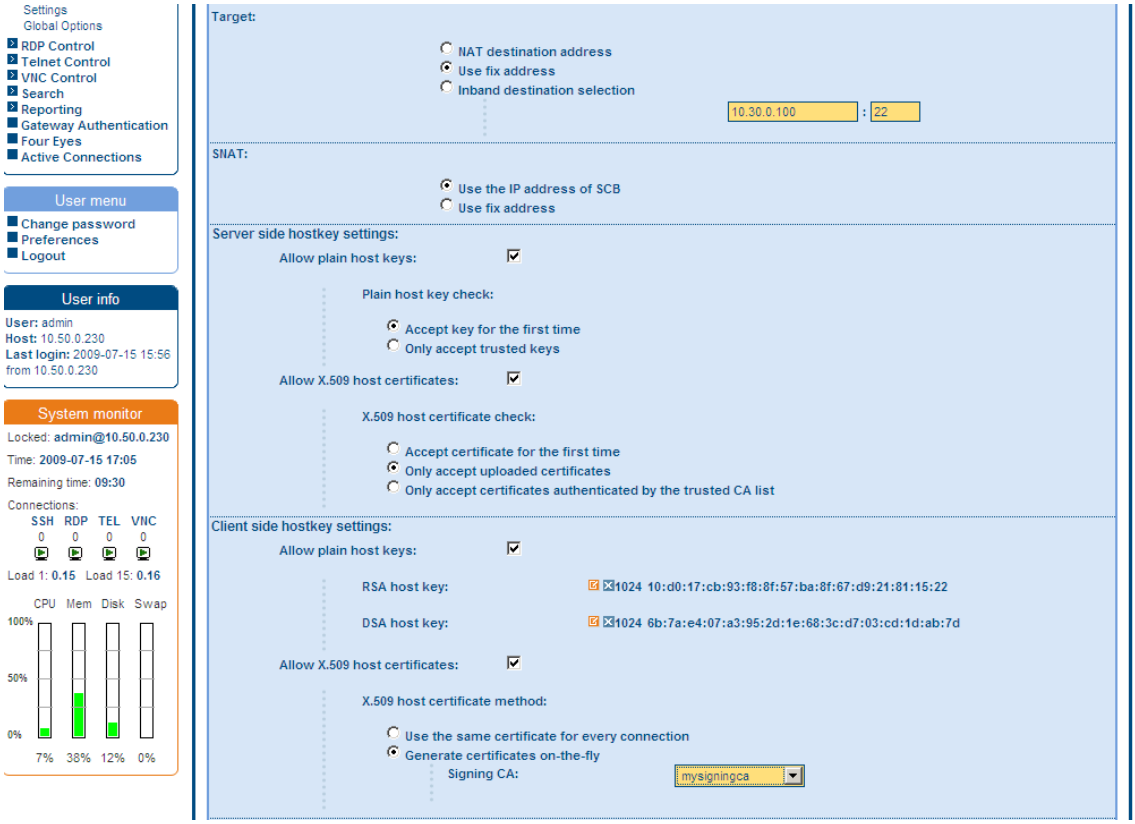


Figure 5.18. Configuring SSH host keys of the connection

Navigate to **SSH Control > Connections** and click to display the details of the connection.

Step 2. To verify the identity of the servers based on their hostkeys, select **Server-side hostkey settings > Allow plain host keys**.

Note
At least one of the **Server-side hostkey settings** options must be enabled.

- Select **Accept the key for the first time** to automatically record the key shown by the server on the first connection. SCB will accept only this key from the server in later connections. This is the default behavior of SCB.
- Select **Only accept trusted keys** if the key of the server is already available on SCB. SCB will accept only the stored key from the server. For further information on setting the host keys of the server, see *Section 5.2.4, Server host keys and certificates (p. 161)*.

Step 3. To verify the identity of the servers based on their X.509 host certificates, select **Server-side hostkey settings > Allow X.509 host certificates**.

Note
At least one of the **Server-side hostkey settings** options must be enabled.

- Select **Accept certificate for the first time** to automatically record the certificate shown by the server on the first connection. SCB will accept only this certificate from the server in later connections.
- Select **Only accept uploaded certificates** if the certificate of the server is already available on SCB. SCB will accept only the stored certificate from the server. for further information on setting the host certificate of the server, see *Section 5.2.4, Server host keys and certificates (p. 161)*.
- Select **Only accept certificates authenticated by the trusted CA list** to verify the host certificate of the server to a CA certificate, and select the Trusted CA list to use in the **Trusted CA list** field. For details on creating CA lists, see *Procedure 5.1.9, Verifying certificates with Certificate Authorities (p. 142)*.

Note
By default, SCB accepts only plain hostkeys, and accepts them for the first time.

Step 4. To set the RSA and DSA host keys that SCB shows to the clients, select **Client side hostkey settings > Allow plain host keys**, and click the icon in the **RSA host key** or the **DSA host key** fields to set the RSA and DSA host keys, respectively. It is possible to upload or paste a key or to generate a new one. Click on the fingerprint to display the public part of the key.

Step 5. To enable SCB to show an X.509 certificate to the clients, select **Client side hostkey settings > Allow X.509 host certificates**.

- To always use the same certificate, select **Use the same certificate for every connection** and upload a private key and a certificate.
- To generate a new certificate for the connection policy (not for every session), select **Generate certificates on-the-fly**, and set the CA to use for signing the certificate in the **Signing CA** field. For details about creating signing CAs, see *Procedure 5.1.10, Signing certificates on-the-fly (p. 143)*.

Step 6. Click **Commit**.

5.2.2. Supported SSH channel types

The available SSH channel types and their functionalities are described below. For a list of supported client applications, see *Section 2.2, Supported protocols and client applications (p. 5)*.

- *Agent*: Forwards the SSH authentication agent from the client to the server.

Note
To perform agent-based authentication on the target server, it is not required to enable the `Agent-forwarding` channel in the Channel Policy used by the connection. The `Agent-forwarding` channel is needed only to establish connections from the target server to other devices and authenticate using the agent running on the client.

- *X11 Forward*: Forwards the graphical X-server session from the server to the client. Enter the address of the client into the **Details > Target address** field to permit X11-forwarding only to the specified clients. Specify IP addresses or networks (in IP address/Netmask format).

Note
Certain client applications send the Target address as a hostname, while others as an IP address. If you are using a mix of different client applications, you might have to duplicate the channel rules and create IP-address and hostname versions of the same rule.

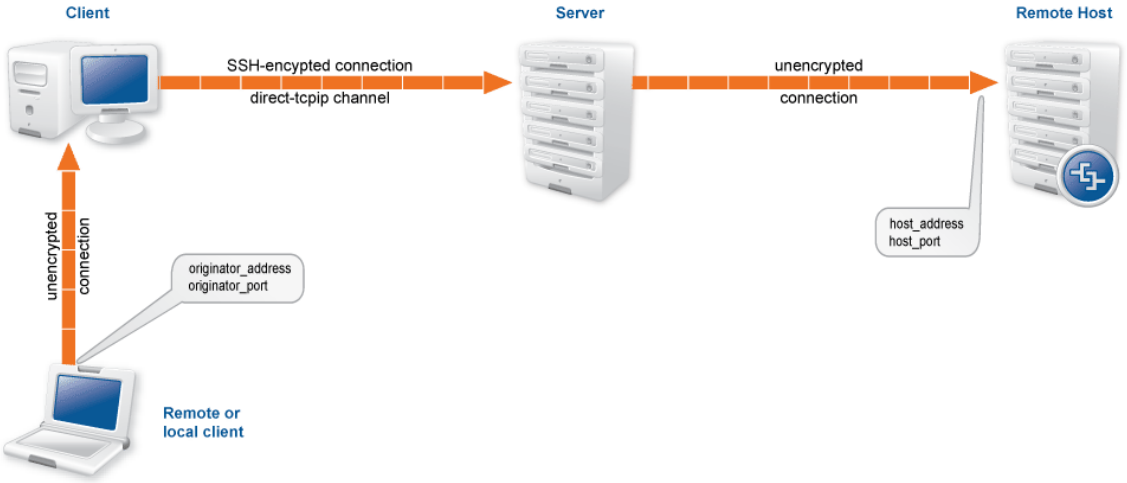
- 

Figure 5.19. Local TCP forwarding

Local Forward: Forwards traffic arriving to a local port of the client to a remote host. To enable forwarding only between selected hosts, enter their IP addresses into the **Details** field. If the **Details** field is empty, local forwarding is enabled without restriction, the client may forward any traffic to the remote host. Enter the source of the forwarded traffic into the **Originator**, the target of the traffic into the **Target** field. Specify IP addresses or networks (in IP address/Netmask format). These parameters are the end-points of the forwarded traffic (that is, the local host that sends data to the remote host), and not the SSH server or the client. For example, to enable forwarding from the `192.168.20.20` host to the remote host `192.168.50.50`, enter `192.168.20.20` into the **Originator**, and `192.168.50.50` into the **Target** field.

Note
Certain client applications send the Originator and Target addresses as hostnames, while others as IP addresses. If you are using a mix of different client applications, you might have to duplicate the channel rules and create IP-address and hostname versions of the same rule.

Warning
Port forwarding across SCB may fail for certain SSH client-server combinations. This happens if within the protocol, the address of the remote host is specified as a hostname during the port-forwarding request (`SSH_MSG_GLOBAL_REQUEST`), but the hostname is resolved to IP address in the channel opening request (`SSH_MSG_CHANNEL_OPEN`. By default, SCB rejects such connections.

To enable these connections, navigate to **SSH Control > Settings**, and disable the **Strict mode** option.

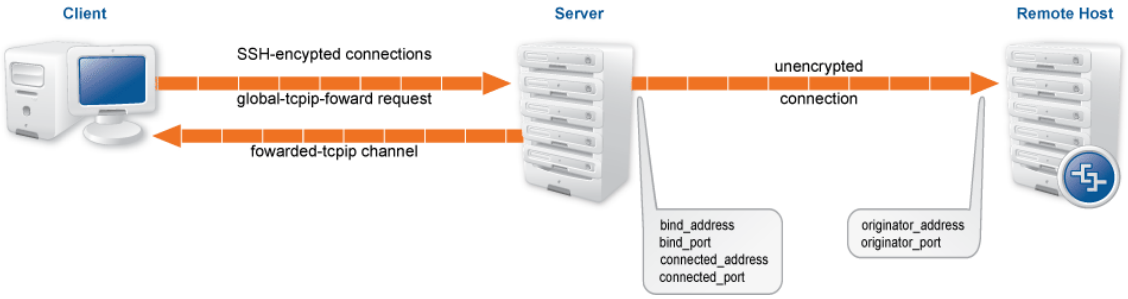


Figure 5.20. Remote TCP forwarding

Remote Forward: Forwards traffic arriving a remote port of the server to the client. To enable forwarding only between selected hosts, enter their IP addresses into the **Details** field. If the **Details** field is empty, remote forwarding is enabled without restriction, the SSH server may forward any traffic to the client. Enter the source of the forwarded traffic into the **Originator**, the target of the traffic into the **Target** field. Specify IP addresses or networks (in IP address/Netmask format). These parameters are the end-points of the forwarded traffic (that is, the remote host that sends data to the client), and not the SSH server. For example, to enable forwarding from the `192.168.20.20` remote host to the client `192.168.50.50`, enter `192.168.20.20` into the **Originator**, and `192.168.50.50` into the **Target** field.

Note
Certain client applications send the Originator and Target addresses as hostnames, while others as IP addresses. If you are using a mix of different client applications, you might have to duplicate the channel rules and create IP-address and hostname versions of the same rule.

Warning
Port forwarding across SCB may fail for certain SSH client-server combinations. This happens if within the protocol, the address of the remote host is specified as a hostname during the port-forwarding request (`SSH_MSG_GLOBAL_REQUEST`), but the hostname is resolved to IP address in the channel opening request (`SSH_MSG_CHANNEL_OPEN`. By default, SCB rejects such connections.

To enable these connections, navigate to **SSH Control > Settings**, and disable the **Strict mode** option.

- *Session Exec*: Execute a remote command (for example `rsync`) without opening a session shell. Enter the permitted command into the **Details** field. Regular expressions may be used to specify the commands.

Warning
Restricting the commands available in **Session Exec** channels does not guarantee that no other commands can be executed. Commands can be renamed, or executed from shell scripts to circumvent such restrictions.

- *Session Exec SCP*: Transfers files using the Secure Copy (SCP) protocol.
  - To make the list of file operations available in the **File operations** column of the **Search > Search** page, navigate to **SSH Control > Channel Policies** and enable the **Log file transfers to database** option. This option is disabled by default.
  - To send the file operations into the system log, enable the **Log file transfers to system** option. This option is disabled by default.

Note
Turning logging on might result in a slight performance penalty. If traffic load slows processes down, disable the option. You will see the file list in the audit player without enabling this option.

- *Session Subsystem*: Use a subsystem. Enter the name of the permitted subsystem into the **Details** field.
- *Session SFTP*: Transfers files using the Secure File Transfer Protocol (SFTP).
  - To make the list of file operations available in the **File operations** column of the **Search > Search** page, select the **Log file transfers to database** option. This option is disabled by default.
  - To send the file operations into the system log, enable the **Log file transfers to system** option. This option is disabled by default.

Note
Turning logging on might result in a slight performance penalty. If traffic load slows processes down, disable the option. You will see the file list in the audit player without enabling this option.

- *Session Shell*: The traditional remote terminal session.

5.2.3. Authentication Policies

An authentication policy is a list of authentication methods that can be used in a connection. Connection definitions refer to an authentication policy to determine how the client can authenticate to the target server. Separate authentication methods can be used on the client and the server-side of the connection.

Figure 5.21. Authentication policies

5.2.3.1. Procedure – Creating a new authentication policy

Purpose:

To create a new authentication policy, follow the steps below:

Steps:

Step 1.

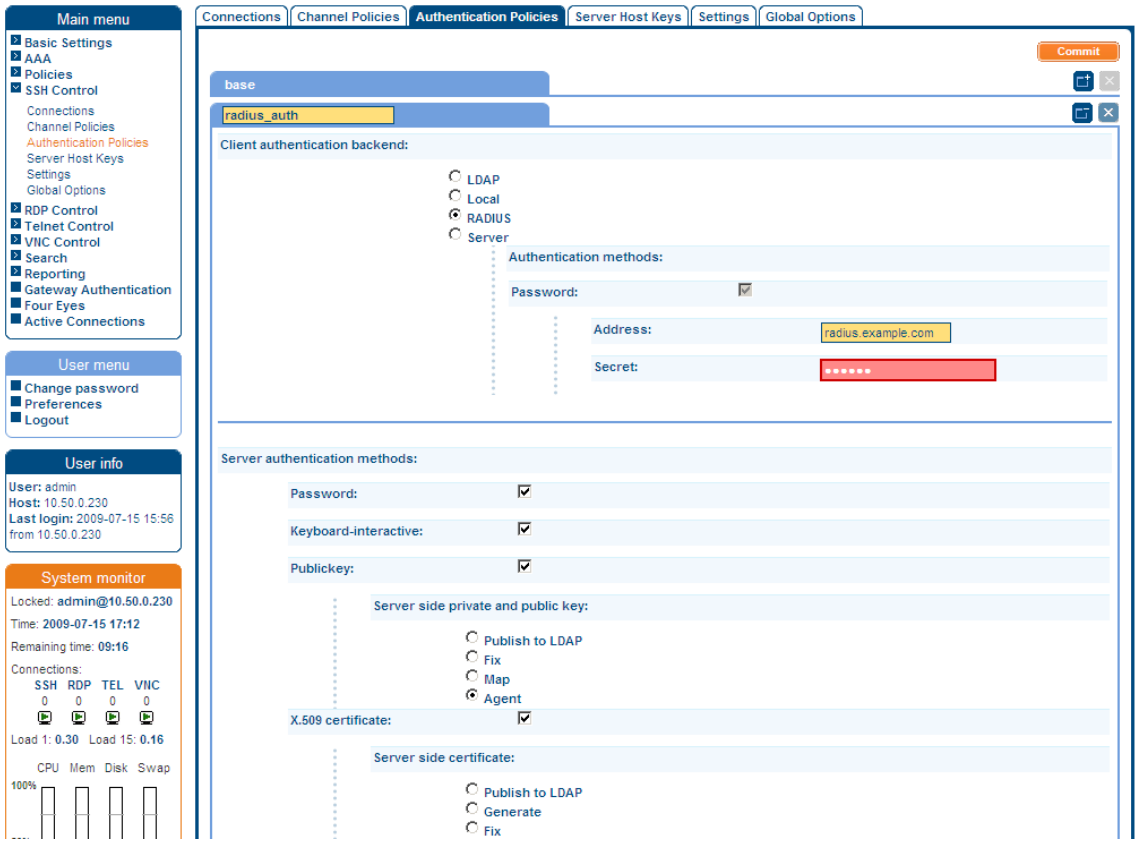


Figure 5.22. Configuring authentication policies

Navigate to **SSH Control > Authentication Policies**, and click ⊞.

Step 2. Enter a name for the policy into the **Name** field.

Step 3. Select the authentication method used on the client-side in the **Client authentication backends** field. For details on the client-side authentication settings, see *Section 5.2.3.2, Client-side authentication settings (p. 156)*.

Step 4. Select the authentication method used on the server-side in the **Server authentication methods** field. For details on the server-side authentication settings, see *Section 5.2.3.3, Server-side authentication settings (p. 158)*.

Step 5. Click **Commit**.

Note
The client-side authentication is independent from the gateway authentication that can be required by the connection policy. Gateway authentication can be used together with authentication policies. In an extreme setting, this would mean that the user has to perform three authentications: a client-side authentication on SCB, a gateway authentication on SCB, and a final authentication on the server.

5.2.3.2. Client-side authentication settings

For the client-side connection (between the client and SCB), the following authentication methods are available:

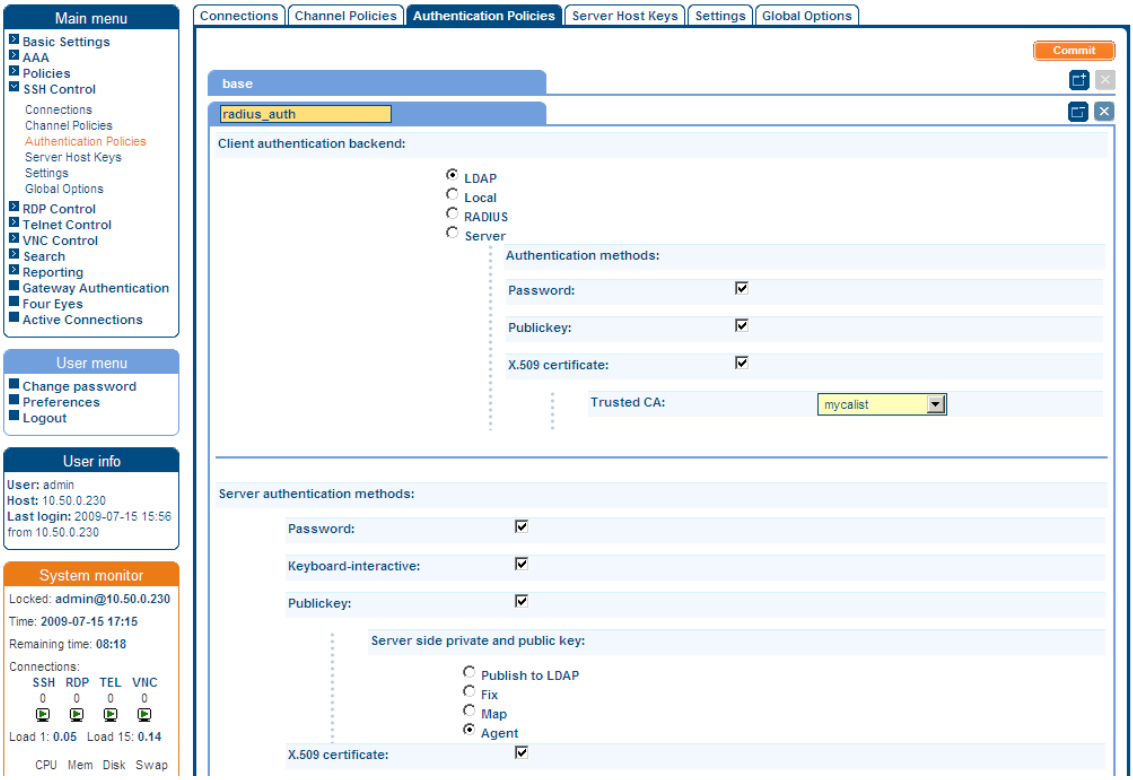


Figure 5.23. Configuring authentication policies

- *LDAP*: SCB will authenticate the client to the LDAP database set in the **LDAP Server** of the connection policy. To use LDAP authentication on the client side, select **Client authentication backends > LDAP**, and select the permitted authentication methods (**Password**, **Public key**, **X.509 certificate**). More than one method can be permitted.

Note

- SCB will authenticate the client-side connection to the LDAP server configured in the connection policy. This is not necessarily the same as the LDAP server used to authenticate the users accessing the SCB web interface.
- The public keys of the users stored in the LDAP database must be in OpenSSH format.

- *Local*: Authenticate the client locally on SCB. For details, see *Procedure 5.2.3.2.1, Local client-side authentication (p. 157)*.
- *RADIUS*: SCB will authenticate the client to the specified RADIUS server. Select **Client authentication backends > RADIUS**, enter the IP address or hostname of the RADIUS server into the **Address** field, and the shared secret of the RADIUS server into the **Shared secret** field. Only password-authentication is supported (including one-time passwords), challenge-response based authentication is not.
- *Server*: Do not perform client-side authentication, the client will authenticate only on the target server.

Warning
Hazard of security breach! If the **Server** authentication option is selected on the client side and SCB is configured to use public-key or certificate based authentication on the server, the user will not be authenticated at all unless gateway authentication is required for the connection.

5.2.3.2.1. Procedure – Local client-side authentication

Purpose:

To perform authentication locally on SCB for client-side connections, complete the following steps:

Note
The users can be authenticated to their public-keys or X.509 certificates uploaded to SCB. It is not possible to perform password-based authentication.

The accounts created to access the SCB web interface cannot be used to authenticate SSH connections.

Steps:

Step 1. Navigate to **SSH Control > Authentication Policies**, and select the authentication policy to modify.

Step 2. Select **Client authentication backends > Local**, and select the permitted authentication methods (**Public key**, **X.509 certificate**).

Step 3. Navigate to the bottom of the policy, and click ⊞.

Step 4. Enter the name of the user into the **Username** field.

Note
If you also use Usermapping policies, enter the username that the client will use on the server side. If you also use gateway authentication, the gateway username can be used as well.

Step 5. Depending on the authentication method selected, upload the public key or X.509 certificate of the client by clicking the icons in the **Client Side (public key/certificate)** field.

Step 6. Repeat Steps 3-5 to add other users as required.

Step 7. Click **Commit**.

5.2.3.3. Server-side authentication settings

For the server-side connection (between SCB and the target server), the following authentication methods are available:

- *Password*: Authentication based on username and password. The server will request a password from the user, even if a password-based authentication was already successful on the client-side.
- *Keyboard Interactive*: Authentication based on exchanging messages between the user and the server. This method includes authentication schemes like S/Key or TIS authentication.
- *Public Key*: Authentication based on public-private encryption keypairs. SCB supports the following public-key authentication scenarios:
  - *Agent*: Allow the client to use agent-forwarding, and use its own keypair on the server-side. If this option is used, SCB requests the client to use its SSH agent to authenticate on the target server.

Note
Agent-based authentication can be combined with other authentication methods in the same Authentication Policy. That way the client can fallback to other methods if there are no keys available on the client or authentication with the provided keys is not successful.

To perform agent-based authentication on the target server, it is not required to enable the `Agent-forwarding` channel in the Channel Policy used by the connection. The `Agent-forwarding` channel is needed only to establish connections from the target server to other devices and authenticate using the agent running on the client.

  - *Fix*: Use the specified private key in the server-side connection. Select **Server authentication methods > Server side private and public key > Public key > Fix**, and click to upload the private key.
  - *Map*: SCB stores the public key of the user and a keypair for every user. This keypair is used in the server-side connection. For details, see *Procedure 5.2.3.4, How to map keys and certificates (p. 159)*.
  - *Publish to LDAP*: SCB generates a keypair, and uses this keypair in the server-side connection. The public key of this keypair is also uploaded to the LDAP database set in the **LDAP Server** of the connection policy. That way the server can authenticate the client to the generated public key stored under the user's username in the LDAP database. Select **Server authentication methods > Server side private and public key > Publish to LDAP**.

Note
SCB generates a keypair for every user of the connection policy, not for every session.

- *X.509 certificate*: Authentication based on X.509 certificates. SCB supports the following certificate-based authentication scenarios:
  - *Agent*: Allow the client to use agent-forwarding, and use its own certificate on the server-side. If this option is used, SCB requests the client to use its SSH agent to authenticate on the target server.

Note
Agent-based authentication can be combined with other authentication methods in the same Authentication Policy. That way the client can fallback to other methods if there are no keys available on the client or authentication with the provided keys is not successful.

To perform agent-based authentication on the target server, it is not required to enable the `Agent-forwarding` channel in the Channel Policy used by the connection. The `Agent-forwarding` channel is needed only to establish connections from the target server to other devices and authenticate using the agent running on the client.

  - *Fix*: Use the specified private key and certificate in the server-side connection. Select **Server authentication methods > Server side certificate > Fix**, and click to upload the private key and the certificate.
  - *Map*: SCB stores an X.509 certificate and the corresponding private key for the user, and uses the stored certificate in the server-side connection. For details, see *Procedure 5.2.3.4, How to map keys and certificates (p. 159)*.
  - *Generate*: SCB generates an X.509 certificate and the corresponding private key for every connection policy, and uses this certificate in the server-side connections. Select **Server authentication methods > Server side certificate > Generate**, and select the certificate authority to use for signing the generated certificates with from the **Signing CA** field. For details on configuring signing CAs, see *Procedure 5.1.10, Signing certificates on-the-fly (p. 143)*.
  - *Publish to LDAP*: SCB generates an X.509 certificate and the corresponding private key for every connection policy, and uses this certificate in the server-side connections. The certificate is also uploaded to the LDAP database set in the **LDAP Server** of the connection policy. That way the server can authenticate the client to the generated certificate stored under the user's username in the LDAP database. Select **Server authentication methods > Server side certificate > Publish to LDAP**, and select the certificate authority to use for signing the generated certificates with from the **Signing CA** field. For details on configuring signing CAs, see *Procedure 5.1.10, Signing certificates on-the-fly (p. 143)*.

Note
SCB generates a certificate for every connection policy, not for every session.

5.2.3.4. Procedure – How to map keys and certificates

Steps:

Step 1. Navigate to **SSH Control > Authentication Policies**, and select the authentication policy to modify.

Step 2.

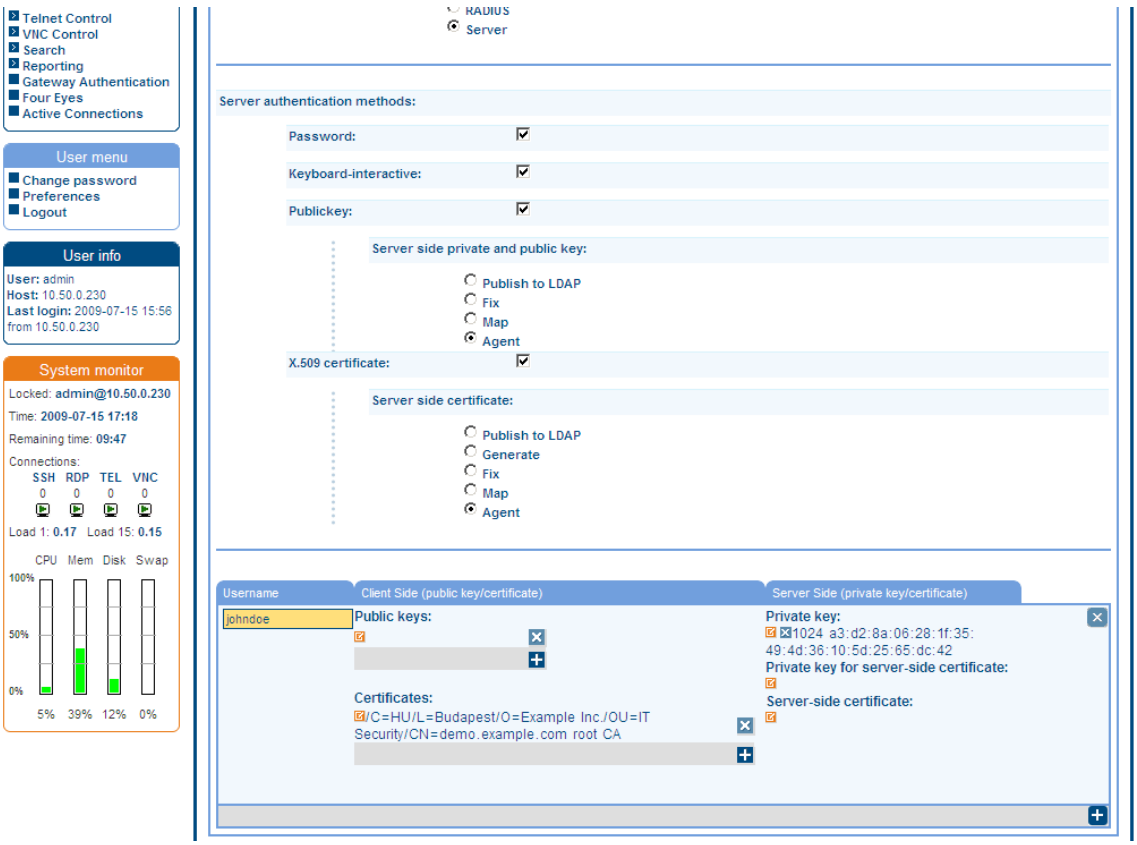


Figure 5.24. Mapping keys and certificates

Navigate to the bottom of the policy, and click ⊞.

Step 3. Enter the name of the user into the **Username** field.

Note
If you also use Usermapping policies, enter the username that the client will use on the server side. If you also use gateway authentication, the gateway username can be used as well.

Step 4.

- If you use public-key based authentication on the client side, click the ⊞ icon in the **Client Side (public key/certificate) > Public keys** field, and upload the public key of the client.
- If you use certificate-based authentication on the client side, click the ⊞ icon in the **Client Side (public key/certificate) > Certificates** field, and upload the certificate of the client.

SCB will verify that a client trying to use the username set in Step 3 is authenticating itself with the private key that corresponds to the uploaded public key or certificate.

Step 5.

- If you use public-key based authentication on the server side, click the ⊞ icon in the **Server Side (public key/certificate) > Private key** field, and upload the private key of the client, or generate a new key.

- If you use certificate-based authentication on the server side, click the icon in the **Server Side (public key/certificate) > Private key for server-side certificate** field, and upload the private key for the certificate of the client. Then upload the certificate that will be used on the server side.

SCB will use the uploaded private key to authenticate on the server if the client is trying to use the username set in Step 3.

Step 6. Repeat Steps 2-5 to add other users as required.

Step 7. Click **Commit**.

5.2.4. Server host keys and certificates

The host keys and X.509 certificates of the trusted servers can be managed using the **Server Host Keys** tab of the **Secure Shell Control** menu item. When a client tries to connect to a server, SCB verifies the host key or the certificate of the server. SCB allows connections only the servers listed on this page, unless the **Accept key for the first time** or the **Accept certificate for the first time** option is enabled in the connection policy.

- To display the stored host keys of a host, enter its IP address into **Host IP address** field and click **Search**. Note that the search requires the exact IP address, and does not support wildcard characters.
- To display the list of stored host keys, click **Show All**. Note that this function does not work if there are more than 250 hosts in the database. In this case, use **Generate CSV** instead.
- To export the entire list into as a comma-separated list, click **Generate CSV**.

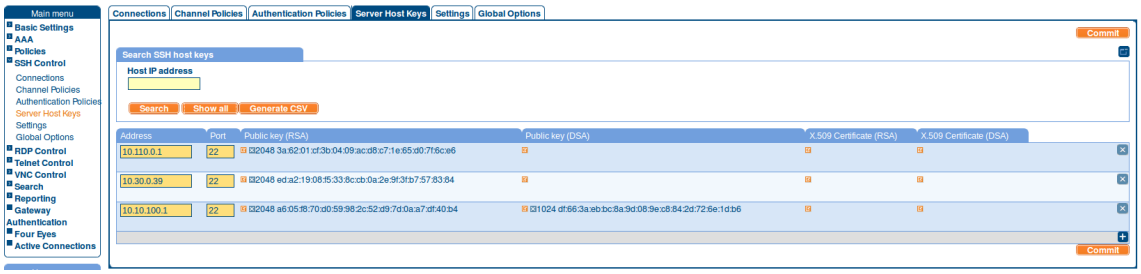


Figure 5.25. Server host keys

5.2.4.1. Procedure – Automatically adding the host keys and host certificates of a server to SCB

Purpose:

The host keys and host certificates of the servers can be added either automatically or manually. To add the host key or certificates automatically, complete the following steps:

Steps:

Step 1. Navigate to the **SSH Control > Connections**.

Step 2. Configure a connection: fill the **From**, **To**, and **Port** fields. If SCB is set to Bastion mode, fill the **Target** field as well.

Note
Enter the IP address of the server into the **To** field if SCB is set to Router, and into the **Target** field if SCB is set to Bastion mode.

Step 3. Click to display the advanced settings and verify that the **Server side hostkey settings > Plain host key check** option is set to **Accept key for the first time**.
If the servers use X.509 certificates, select **Allow X.509 host certificates**, and verify that the **X.509 host certificate check** option is set to **Accept certificate for the first time**.

Click **Commit**.

Step 4. Initiate an SSH connection from the client to the server. SCB will automatically record the host key of the server — the server's IP address and the host key will be listed on the **SSH Control > Server Host Keys** page.

5.2.4.2. Procedure – Manually adding the host key or host certificate of a server

Purpose:

To add the host key or host certificate manually, complete the following steps:

Steps:

Step 1. Navigate to the **SSH Control > Server Host Keys** and click .

Step 2. Enter the IP address and port of the server into the **Address** and **Port** fields.

Step 3. To set the host key of the server, complete the following steps:

Step a.

- To add the RSA fingerprint of the server, click in the **Public key (RSA)** field.
- To add the DSA fingerprint of the server, click in the **Public key (DSA)** field.

A popup window is displayed.

Step b.

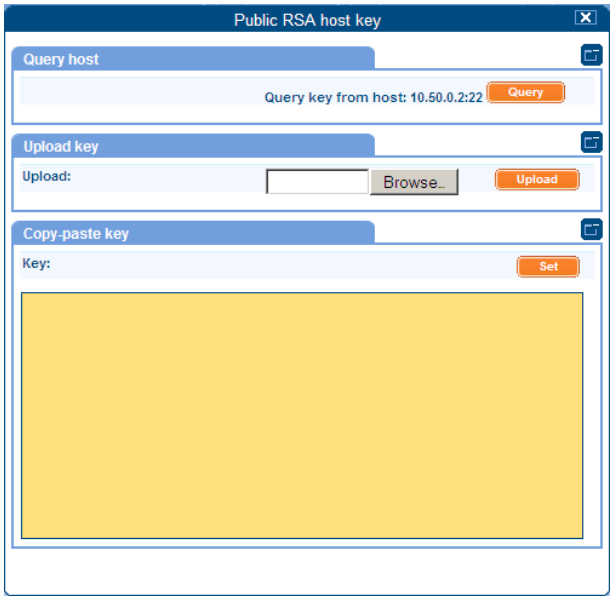


Figure 5.26. Uploading server host keys

- Select **Query** to retrieve the host key from the server.
- To upload the host key manually, select **Browse**, select the file, and click **Upload**. Optionally, you can also paste the key into the **Copy-paste key** section and select **Upload**.
  Select **Close** to close the window.

Step 4. To set the host certificates of the server, complete the following steps:

Step a.

- To add the RSA certificate of the server, click in the **X.509 certificate (RSA)** field.
- To add the DSA certificate of the server, click in the **X.509 certificate (DSA)** field.

A popup window is displayed.

Step b. To upload the host key manually, select **Browse**, select the file, and click **Upload**. Optionally, you can also paste the key into the **Copy-paste key** section and select **Upload**.
Select **Close** to close the window.

Step 5. Click **Commit**.

5.2.5. Procedure – Creating and editing protocol-level SSH settings

Purpose:

SSH settings determine the parameters of the connection on the protocol level, including when the server-side connection is built, as well as the timeout value and greeting message of the connection. The following parameters determine which algorithms are used in the connections, and can be set independently for the client and the server

side: key exchange, host key, cipher, MAC, and compression algorithms. Complete the following procedure to create a new SSH settings profile or edit an existing one:

Warning
Modifying the SSH settings is recommended only to advanced users. Do not modify these settings unless you exactly know what you are doing.

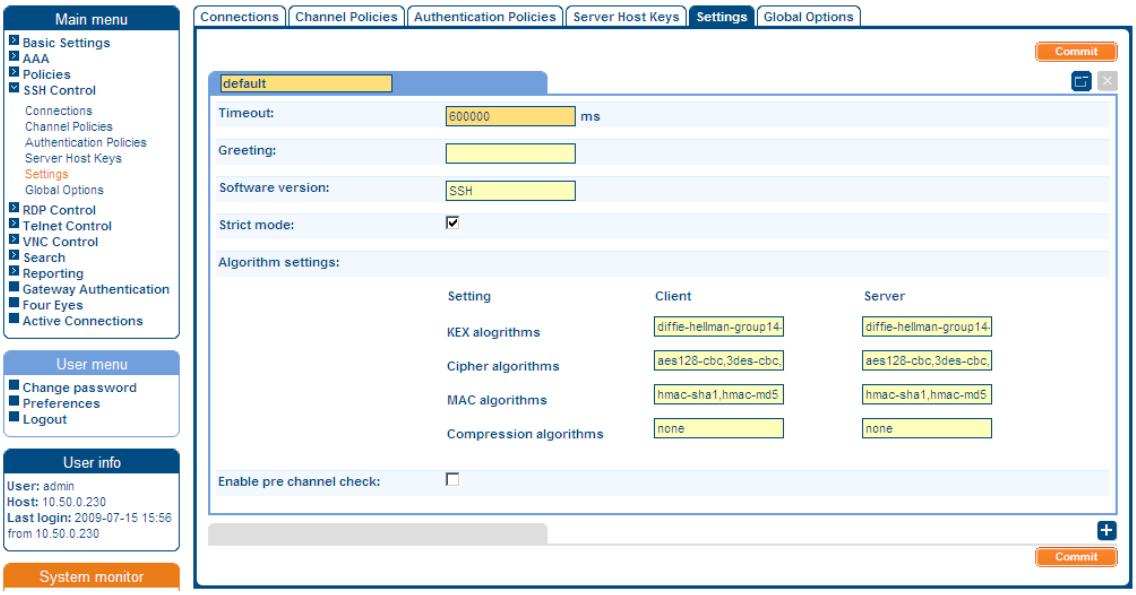


Figure 5.27. SSH settings

Steps:

Step 1. Navigate to the **SSH Control > Settings** and click ⊞ to create an SSH setting profile. Enter a name for the profile into the **SSH Settings** field (for example `strongencryption`).

Step 2. Click ⊞ to display the parameters of the SSH connection.

Step 3. Modify the parameters as needed. The parameters can be set independently for the client- and the server-side connection. For a list of the available parameters, see *Section 1.2, Configuring encryption parameters (p. 248)*.

Step 4. To check the protocol-level parameters of the connections very strictly, select the **Strict mode** option. This option is enabled by default. When this option is enabled, SCB will reject connections that use unrealistic parameters (for example terminals of thousand by thousand characters) and port-forwarding connections where the address in the port-forwarding request and the channel-opening request does not match. Note that this option can interfere with certain client or server applications.

Step 5. Before establishing the server-side connection, SCB can evaluate the connection and channel policies to determine if the connection might be permitted at all, for example it is not denied by a Time Policy. To enable this function, select the **Enable pre channel check** option. That way SCB establishes the server-side connection only if the evaluated policies permit the client to access the server.

Step 6. Click **Commit**.

Step 7. Select this settings profile in the **SSH Settings** field of your connections.

5.3. RDP-specific settings

The following sections describe configuration settings available only for the RDP protocol. Use the following policies to control who, when, and how can access the RDP connection.

- *Channel Policy*: The channel policy determines which RDP channels (for example clipboard, file-sharing, and so on) can be used in the connection, and whether they are audited or not. The different channels may be available only under certain restrictions, as set in the channel policy. For details, see *Procedure 5.1.4, Creating and editing channel policies (p. 126)*.

- *RDP settings*: RDP settings determine the parameters of the connection on the protocol level, including timeout value, display parameters, and the version of RDP permitted. For details, see *Procedure 5.3.2, Creating and editing protocol-level RDP settings (p. 166)*.

- *Domain membership*: When using Network Layer Authentication (CredSSP) SCB must be a member of the domain. For details, see *Procedure 5.3.3, Joining SCB into a domain (p. 168)*.
- *SSL-encrypted connections*: For details on how to setup SSL-encrypted RDP connections, see *Procedure 5.3.4, Using SSL-encrypted RDP connections (p. 169)* and *Procedure 5.3.5, Verifying the certificate of the RDP server in encrypted connections (p. 170)*.

5.3.1. Supported RDP channel types

The available RDP channel types and their functionalities are described below. For a list of supported client applications, see *Section 2.2, Supported protocols and client applications (p. 5)*.

- *Drawing*: Enables access to the server's desktop (screen). This channel must be enabled for RDP to work.
- *Clipboard*: Enable access to the server's clipboard: the clipboard of the remote desktop can be pasted into local applications (and vice-versa). Note that SCB can audit the clipboard channel, but the Audit Player currently cannot search or display its contents.
- *Redirects*: Enables access to every device redirections available in RDP, like file-sharing, printer sharing, device (for example CD-ROM) sharing, and so on. To enable only specific types of redirections, use the following channels:
  - *Serial redirect*: Enables access to serial-port redirections.
  - *Parallel redirect*: Enables access to parallel-port redirections.
  - *Printer redirect*: Enables access to shared printers.
  - *Disk redirect*: Enables access to shared disk drives.
  - *SCard redirect*: Enables access to shared SCard devices.

  To permit only specific redirections, enter the unique name of the redirection into the **Details** field. For example, if you want to enable access only to the shared disk drive C:\, enable the **Disk redirect** channel and enter `C:\` into the **Details** field. Note that the name of the device comes from the device itself, so it is case sensitive, and may not always be reliable from a security point of view.

- *Sound*: Enable access to the sound device of the server.

- *Custom*: Applications can open custom channels to the clients connecting remotely to the server. Enabling the **Custom** channel allows the clients to access all of these custom channels. To permit only specific channels, enter the unique names of the channel into the **Details** field.
- *Seamless*: Enable seamless channels that run a single application on the RDP server, instead of accessing the entire desktop.
- *Dynamic virtual channel*: Enable the server to open channels back to the client dynamically. To restrict which dynamic channels are permitted, select **Channel details >** ⊞, and enter the name of the permitted channel.

5.3.2. Procedure – Creating and editing protocol-level RDP settings

Purpose:

RDP settings determine the parameters of the connection on the protocol level, including timeout value, the version of RDP permitted in the connection, and display parameters. Complete the following procedure to create a new RDP settings profile or edit an existing one:

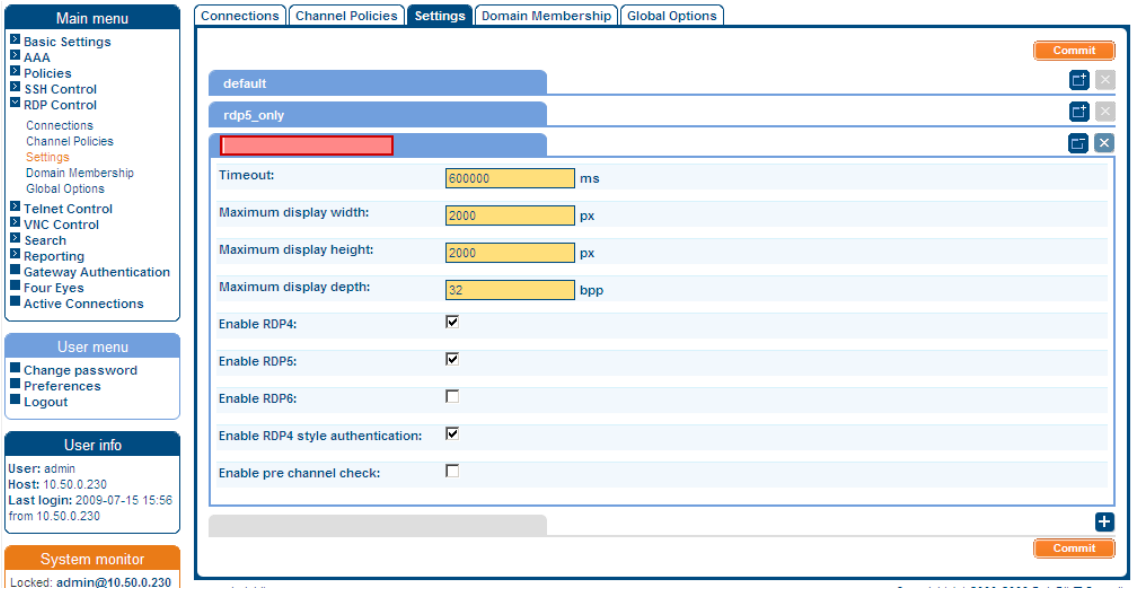


Figure 5.28. RDP settings

Warning
Modifying the RDP settings is recommended only to advanced users. Do not modify these settings unless you exactly know what you are doing.

Steps:

Step 1. Navigate to **RDP Control > Settings** and click ⊞ to create an RDP setting profile. Enter a name for the profile into the **RDP Settings** field (for example `rdp5only`).

Step 2. Click ⊞ to display the parameters of the RDP connection.

Step 3. Modify the parameters as needed. The following parameters are available:

- *Timeout*: Timeout value for the connection in milliseconds. Negative numbers (for example `-1`) set the timeout to unlimited.
- *Maximum display width*: The maximum allowed width of the remote desktop in pixels (for example `1024`).
- *Maximum display height*: The maximum allowed height of the remote desktop in pixels (for example `768`).
- *Maximum display depth*: The maximum allowed color depth the remote desktop in bits (for example `24`). The following values are valid: 8, 15, 16, 24.

Warning
Using 32-bit color depth is not supported.

Warning
Certain Windows versions do not support 24-bit color depth. In this case, those versions can only be displayed in 16-bit color depth. SCB automatically changes its settings to 16-bit.

- *Enable RDP4*: Select this option to enable the version 4 of the Remote Desktop Protocol.
- *Enable RDP5*: Select this option to enable the version 5 of the Remote Desktop Protocol.
- *Enable RDP6*: Select this option to enable the use of Credential Security Service Provider (CredSSP, also called Network Layer Authentication). Note that: SSL-encrypted connections (even in RDP6) do not require this option, it is only needed for Network Layer Authentication. If you enable this option, you also have to configure SCB to join your domain. For details, see *Procedure 5.3.3, Joining SCB into a domain (p. 168)*. Also note that Smartcard authentication cannot be used when this option is enabled.

Warning
To access hosts running Windows 2008 Server R2 using RDP6, select the *Enable RDP4 style authentication* option as well.

- *Enable RDP4 style authentication*: Select this option to enable RDP4 authentication within the RDP5 protocol. This might be needed for compatibility reasons with certain client applications.
- *Enable pre channel check*: Select this option to evaluate the connection and channel policies before establishing the server-side connection. That way if the connection is not permitted at all SCB does not establish the server-side connection.

Step 4. Click **Commit**.

Step 5. Select this settings profile in the **RDP Settings** field of your connections.

5.3.3. Procedure – Joining SCB into a domain

Purpose:

Joining a domain is required when using Credential Security Service Provider (CredSSP, also called Network Layer Authentication).

Steps:

Step 1. Navigate to **RDP Control > Domain membership**.

Step 2. Enter the name of the domain (for example *mydomain*) into the **Domain** field.

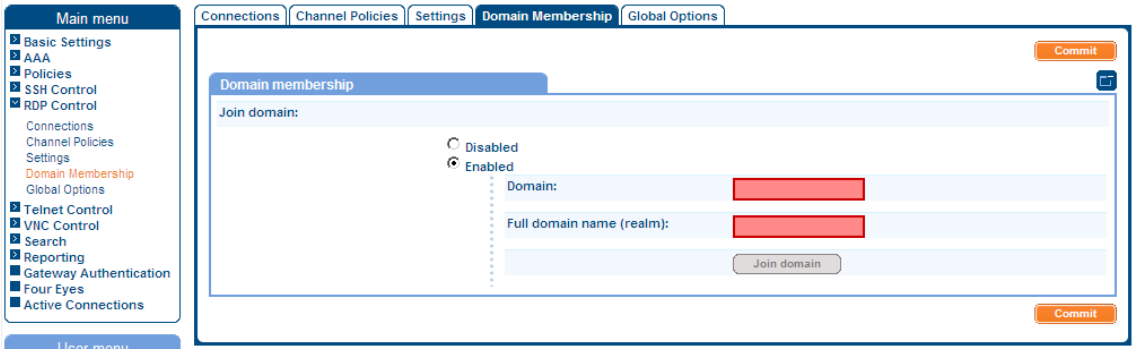


Figure 5.29. Joining a domain

Step 3. Enter the name of the realm (for example *mydomain.example.com*) into the **Full domain name** field.

Note
Ensure that your DNS settings are correct and that the full domain name can be resolved from SCB. To check this, navigate to **Basic Settings > Troubleshooting > Ping**, enter the full domain name into the **Hostname** field, and select **Ping host**.

Step 4. Click **Join domain**. A popup window is displayed.

Step 5. SCB requires an account to your domain to be able to join the domain. Enter the name of the user into the **Username** field, and the corresponding password into the **Password** field.
Optionally, you can enter the name of your domain controller into the **Domain controller** field. If you leave this field blank, SCB will try to find the domain controller automatically.

Note
Ensure that your DNS settings are correct and that the hostname of the domain controller can be resolved from SCB. To check this, navigate to **Basic Settings > Troubleshooting > Ping**, enter the name of the domain controller into the **Hostname** field, and select **Ping host**.

Step 6. Click **Join domain**.

5.3.4. Procedure – Using SSL-encrypted RDP connections

Purpose:

RDP5 and RDP6 connections may use SSL encryption to encrypt the communication between the client and the server. If both the client and the server supports SSL encryption, the connection will be encrypted.

Note
Using Network Layer Authentication (CredSSP) automatically uses SSL-encryption, but does not require a signing CA.

To enable SSL encryption, the **Enable RDP5** option must be enabled in the protocol settings of the connection. In the default setting, this is enabled. To enable Network Layer Authentication (CredSSP) the **Enable RDP6** option must be enabled in the protocol settings of the connection. In the default setting, this is not enabled. For details, see *Procedure 5.3.2, Creating and editing protocol-level RDP settings (p. 166)*.

When the RDP connection is SSL-encrypted, SCB has to show a certificate to the client. To set the certificate authority used to sign these certificates, complete the following steps.

Steps:

Step 1. Create a certificate authority that will be used to sign the certificates that SCB shows to the client. For details, see *Procedure 5.1.10, Signing certificates on-the-fly (p. 143)*.

Step 2. Navigate to **RDP Control > Connections** and select the connection policy to modify.

Step 3.

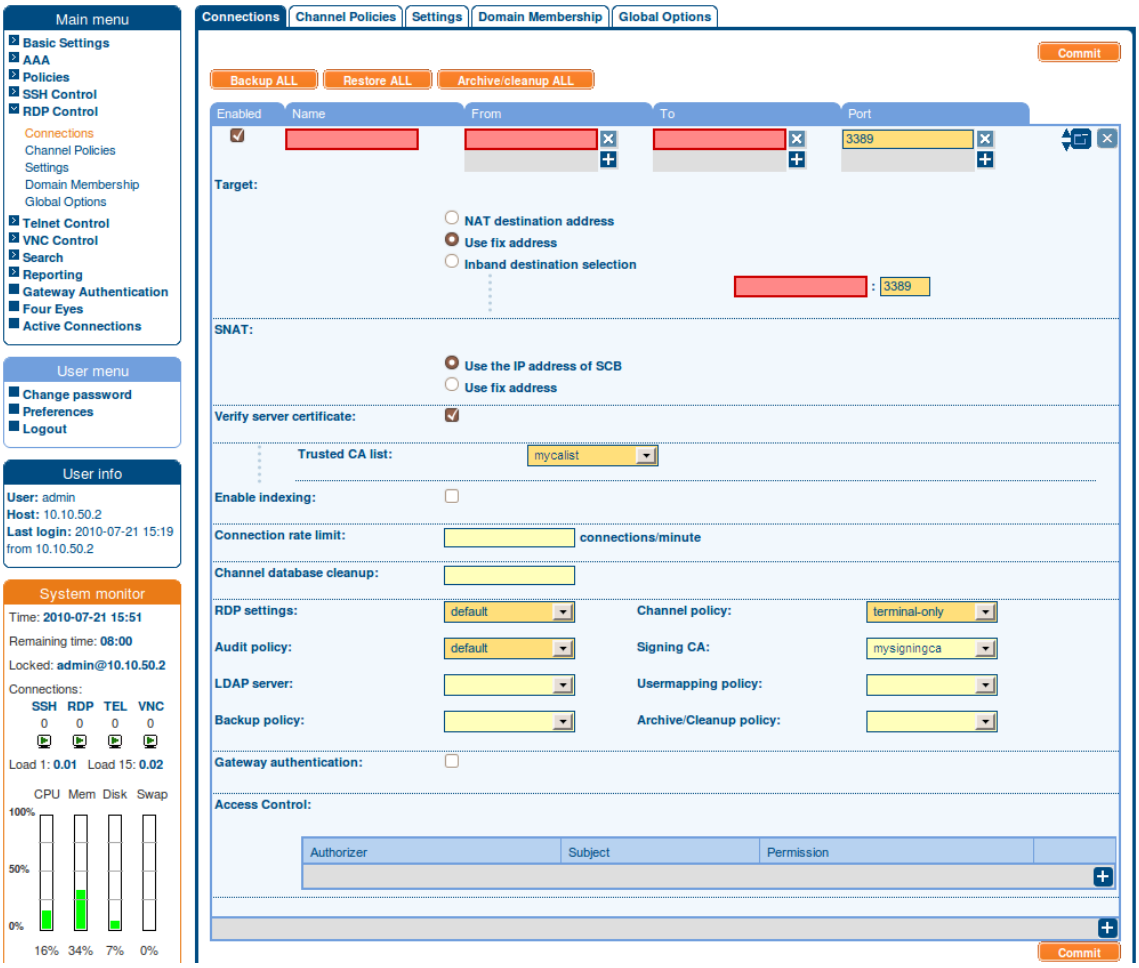


Figure 5.30. Using SSL-encryption in RDP connections

In the **Signing CA** field, select the certificate authority to use.

Warning
SSL-encrypted RDP connections will be automatically rejected if no signing CA is selected.

Note
Connections using Network Layer Authentication (CredSSP) do not need a signing CA, because they use self-signed certificates that are created automatically.

Step 4. Click **Commit**.

5.3.5. Procedure – Verifying the certificate of the RDP server in encrypted connections

Purpose:

By default, SCB accepts any certificate shown by the server. To accept only verified certificates, complete the following steps:

Steps:

Step 1. Create a list of trusted CA certificates that will be used to verify the certificate of the server. For details, see *Procedure 5.1.9, Verifying certificates with Certificate Authorities (p. 142)*.

Step 2. Navigate to **RDP Control > Connections** and select the connection policy to modify.

Step 3.

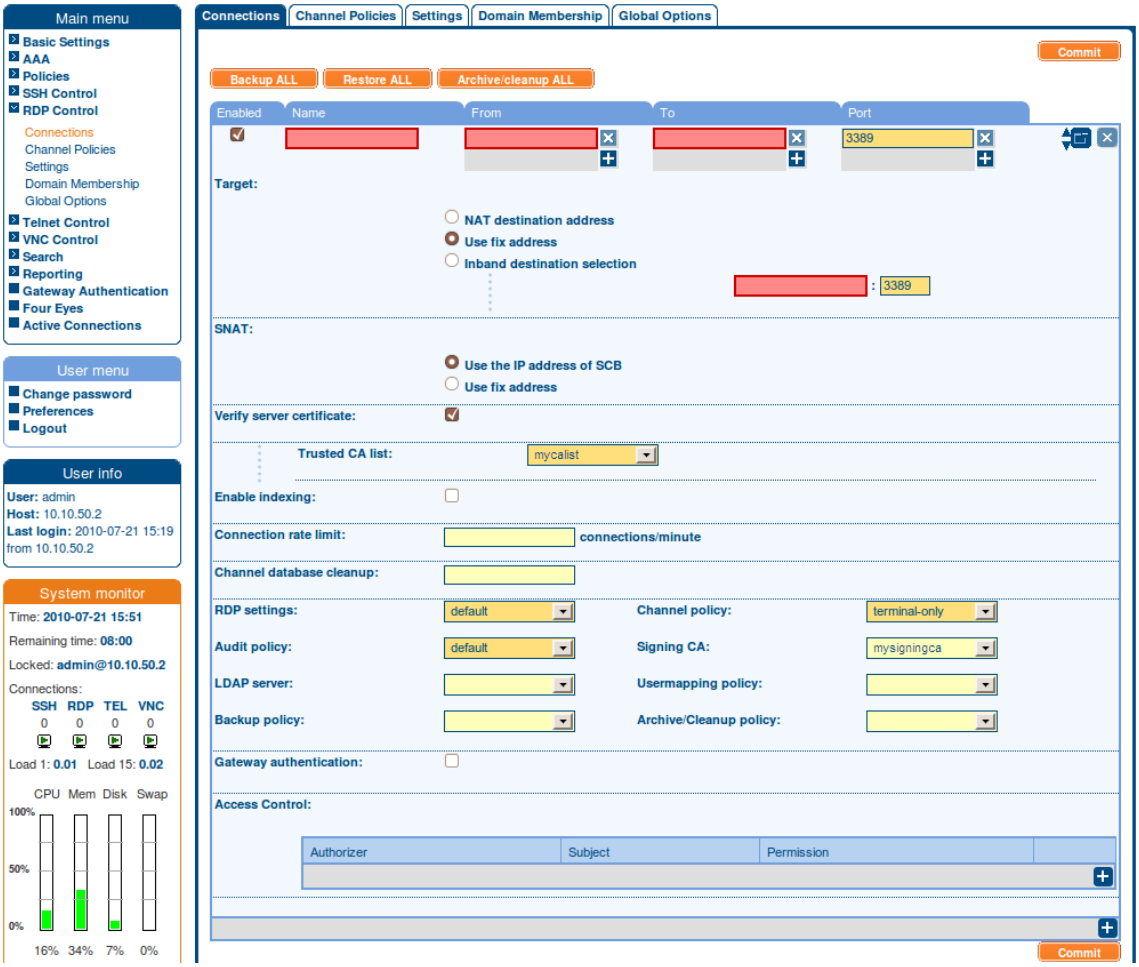


Figure 5.31. Using SSL-encryption in RDP connections

Select **Verify server certificate**.

Step 4. Select the CA list to use for verifying the certificate of the server from the **Trusted CA list** field.

Step 5. Click **Commit**.

5.4. Telnet-specific settings

The following sections describe configuration settings available only for the Telnet protocol. Use the following policies to control who, when, and how can access the Telnet connection. For a list of supported client applications, see *Section 2.2, Supported protocols and client applications (p. 5)*.

- *Telnet settings*: Telnet settings determine the parameters of the connection on the protocol level, including timeout value, and so on. For details, see *Procedure 5.4.1, Creating and editing protocol-level Telnet settings (p. 172)*.
- *User lists in Channel Policies*: User lists affect Telnet connections only when they are used together with Gateway Authentication. For details, see *Section 8.2, Configuring gateway authentication (p. 210)*.

5.4.1. Procedure – Creating and editing protocol-level Telnet settings

Procedure:

Telnet settings determine the parameters of the connection on the protocol level, including timeout value, and so on. Complete the following procedure to create a new Telnet settings profile or edit an existing one:

Warning
Modifying the Telnet settings is recommended only to advanced users. Do not modify these settings unless you exactly know what you are doing.

Steps:

Step 1. Navigate to the **Settings** tab of the **Telnet Control** menu item and click ⊕ to create a Telnet setting profile. Enter a name for the profile into the **Telnet Settings** field (for example `telnet_special`).

Step 2. Click ⊡ to display the parameters of the Telnet connection.

Step 3. Modify the parameters as needed. The following parameters are available:

- *Timeout*: Timeout value for the connection in milliseconds. Negative numbers (for example `-1`) set the timeout to unlimited.
- *Enable pre channel check*: Select this option to evaluate the connection and channel policies before establishing the server-side connection. That way if the connection is not permitted at all SCB does not establish the server-side connection.

Step 4. Click **Commit**.

Step 5. Select this settings profile in the **Telnet Settings** field of your connections.

5.5. VNC-specific settings

The following sections describe configuration settings available only for the Virtual Networking (VNC) protocol. Use the following policies to control who, when, and how can access the VNC connections. For a list of supported client applications, see *Section 2.2, Supported protocols and client applications (p. 5)*.

- *VNC settings*: VNC settings determine the parameters of the connection on the protocol level, including timeout value, and so on. For details, see *Procedure 5.5.1, Creating and editing protocol-level VNC settings (p. 173)*.

- *User lists in Channel Policies*: User lists affect VNC connections only when they are used together with Gateway Authentication. For details, see *Section 8.2, Configuring gateway authentication (p. 210)*.

5.5.1. Procedure – Creating and editing protocol-level VNC settings

Purpose:

VNC settings determine the parameters of the connection on the protocol level, including timeout value, and so on. Complete the following procedure to create a new VNC settings profile or edit an existing one:

Warning
Modifying the VNC settings is recommended only to advanced users. Do not modify these settings unless you exactly know what you are doing.

Steps:

Step 1. Navigate to **VNC Control > Settings** and click ⊞ to create a VNC setting profile. Enter a name for the profile into the **VNC Settings** field (for example `vnc_special`).

Step 2. Modify the parameters as needed. The following parameters are available:

- *Timeout*: Timeout value for the connection in milliseconds. Negative numbers (for example `-1`) set the timeout to unlimited.
- *Enable pre channel check*: Select this option to evaluate the connection and channel policies before establishing the server-side connection. That way if the connection is not permitted at all SCB does not establish the server-side connection.

Step 3. Click **Commit**.

Step 4. Select this settings profile in the **VNC Settings** field of your connections.

5.6. VMware View connections

The following sections describe how to use SCB to control and audit VMware View connections. When using SCB to control and audit VMware View connections, the following requirements and restrictions apply:

- Only connections using the Remote Desktop (RDP) display protocol are supported. Connections using the PCoIP or HP Remote Graphics Software display protocols are not supported.
- Both direct connections and tunnel connections are supported.
- The VMware View connections must pass SCB directly; it is best if SCB is deployed in Router or Bridge mode directly before the Virtual Desktops accessed with VMware View.
  Deploying SCB that way has the advantage of auditing connections even if the clients access the Virtual Desktops directly, without using a View Connection Server.

Note
Using Bastion mode is also possible if the VMware View traffic is routed to SCB with an external device (for example, a firewall).

SCB treats VMware View connections that satisfy these criteria as common RDP connections. All the features of SCB that are available for RDP connections can be used with VMware View connections as well, for example, 4-eyes authorization, auditing and replaying, indexing the recorded audit trails, and so on. For details on RPD-specific settings, see *Section 5.3, RDP-specific settings (p. 165)*.

Chapter 6. Browsing log messages and SCB reports

This section describes how to browse the various types of log messages and audit trails on SCB and exactly what kind of information do they contain.

- For the general use of the search interfaces, see *Section 6.1, Using the search interface (p. 175)*. Some of the search interfaces has certain special features, these are described in their respective sections.
- *Accounting information about the configuration changes performed on the SCB web interface*: Shows the activity of the SCB users and administrators. Available at **AAA > Accounting**. For the list of displayed parameters, see *Section 6.2, Changelogs of SCB (p. 178)*.
- *Recorded audit trails*: Audit trails stored on SCB or archived to a remote server. To browse these audit trails, select **Search > Search**. For the list of displayed parameters, see *Section 6.4, The SCB connection database (p. 181)*.
- *SCB reports*: PDF reports about the configuration changes, system health parameters, and other activities of SCB. Available at **Reporting > Reports**. For the list of displayed parameters, see *Section 6.3, Reports (p. 179)*.
- *Custom reports*: PDF reports about the custom searches performed in the audit trails, available at **Reporting > Reports**. For the list of displayed parameters, see *Procedure 6.5, Configuring custom reports (p. 185)*.
- For details about using and monitoring the indexing of audit trails, see *Section 6.6, Monitoring the status of AP indexing services (p. 188)* and *Procedure 6.7, Configuring full-text indexing of audit trails (p. 189)*.

6.1. Using the search interface

SCB has a uniform interface for browsing SCB configuration changes, reports, and audit trails. This search interface consists of two main parts: a calendar bar and a table.

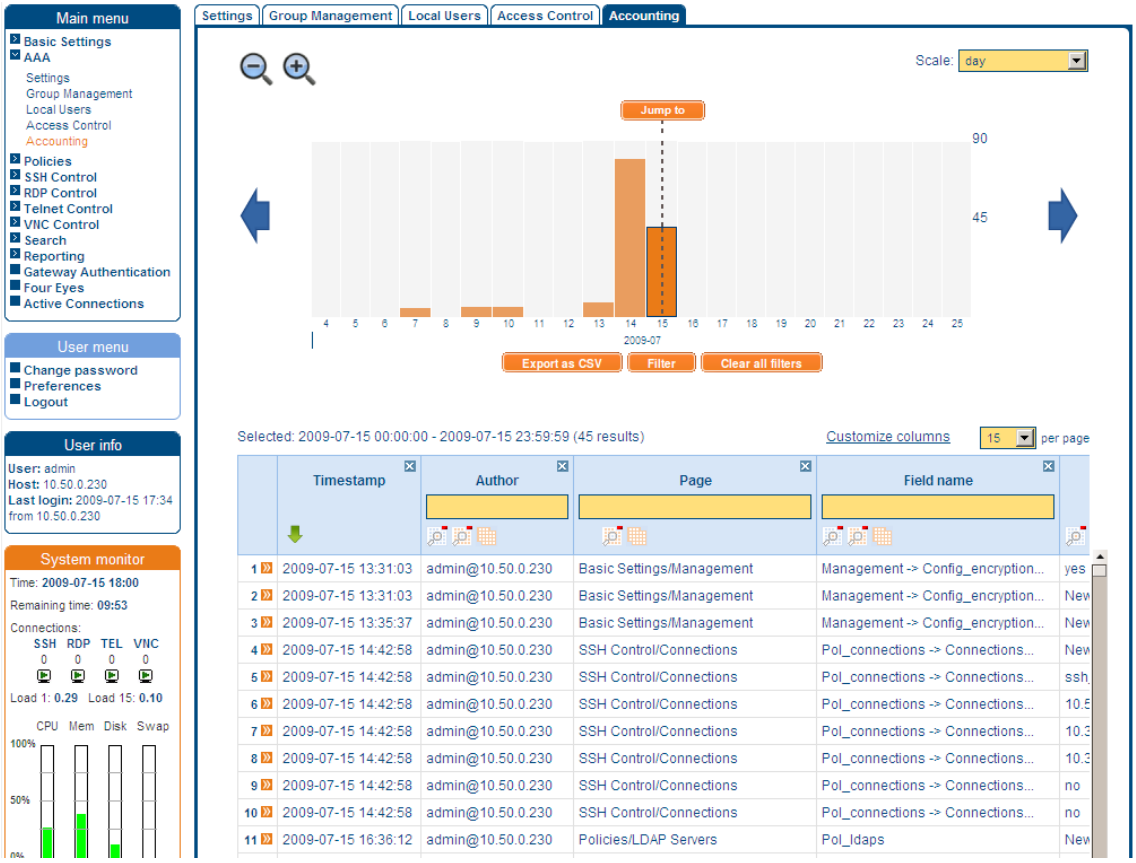


Figure 6.1. Using the search interface

The calendar bar displays the number of log messages in the selected interval. Use the , icons to zoom, and the arrows to display the previous or the next intervals. To explicitly select a date, select **Jump to** and set the date in the calendar. To select the length of the displayed interval use the **Scale** option.

Hovering the mouse above a calendar bar displays the number of entries and the start and end date of the period that the bar represents. Click a calendar bar to display the entries of that period in the table. Use Shift+Click to select multiple calendar bars. The **Selected** field shows the starting and ending date of the period listed in the table.

6.1.1. Procedure – Displaying information about connections

Purpose:

SCB records several parameters of every log message. To select the data displayed, complete the following steps:

Steps:

Step 1. Navigate to the database you want to browse, for example **AAA > Accounting**.

Step 2. Click **Customize Columns**. A popup window containing the list of visible and available columns and dynamic columns is displayed.

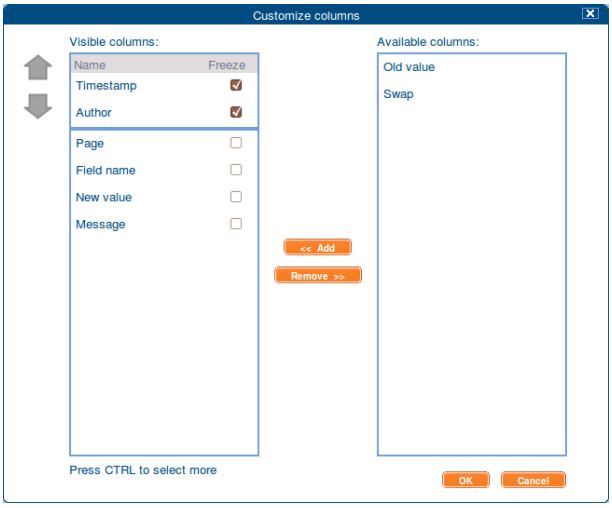


Figure 6.2. Displaying search information

Step 3. The displayed parameters are enlisted in the **Visible columns** field. All other available parameters are enlisted in the **Available columns** field.

- To add parameters to the **Visible columns** field, select the desired parameter(s) and click **Add**.
- To remove parameters from the **Visible columns** field, select the desired parameter(s) and click **Remove**.
- To freeze columns (to make them permanently visible, even when scrolling), enable the **Freeze** option next to the desired parameter.

Note
To select multiple parameters, press **Ctrl** while clicking the items.

Step 4. Click **OK**. The selected information is displayed.

If data is too long to fit on one line, it is automatically wrapped and only the first line is displayed. To expand a row, click ⊞. To shrink the row back to its original size, click ⊟. To expand/shrink all rows, click the respective button on the header of the table. The rows can also be expanded/shrunk by double clicking on the respective row.

The table can be filtered for any parameter, or a combination of parameters. To filter the list, enter the filter expression into the text box and press **Enter**, or click on an entry in the table. For example, to display only changes performed by a specific user, enter the username into the text box and press **Enter** — or just click on the specific username in the table.

Note
When you use filters, the calendar bar displays the statistics of the filtered results.

Filtering displays also partial matches: for example filtering **Author** for `adm` will display all changes performed by users whose username contains the `adm` string.

To save the table of search results as a file, click **Export as CSV**. This saves the table as a text file containing comma-separated values. Note that if an error occurs when exporting the data, the exported CSV file will include a line (usually as the last line of the file) starting with a zero and the details of the problem, for example `0;description_of_the_error`.

To save the audit trail of a session, click the icon in the **Audit trail** column.

To restore the original table, click **Clear all filters**.

Tip
Use the **SSH**, **RDP**, and **Telnet** buttons to quickly filter the list for a single protocol.

6.2. Changelogs of SCB

SCB automatically records the activity of its users and administrators. These activities are displayed at **AAA > Accounting**. The following information is available:

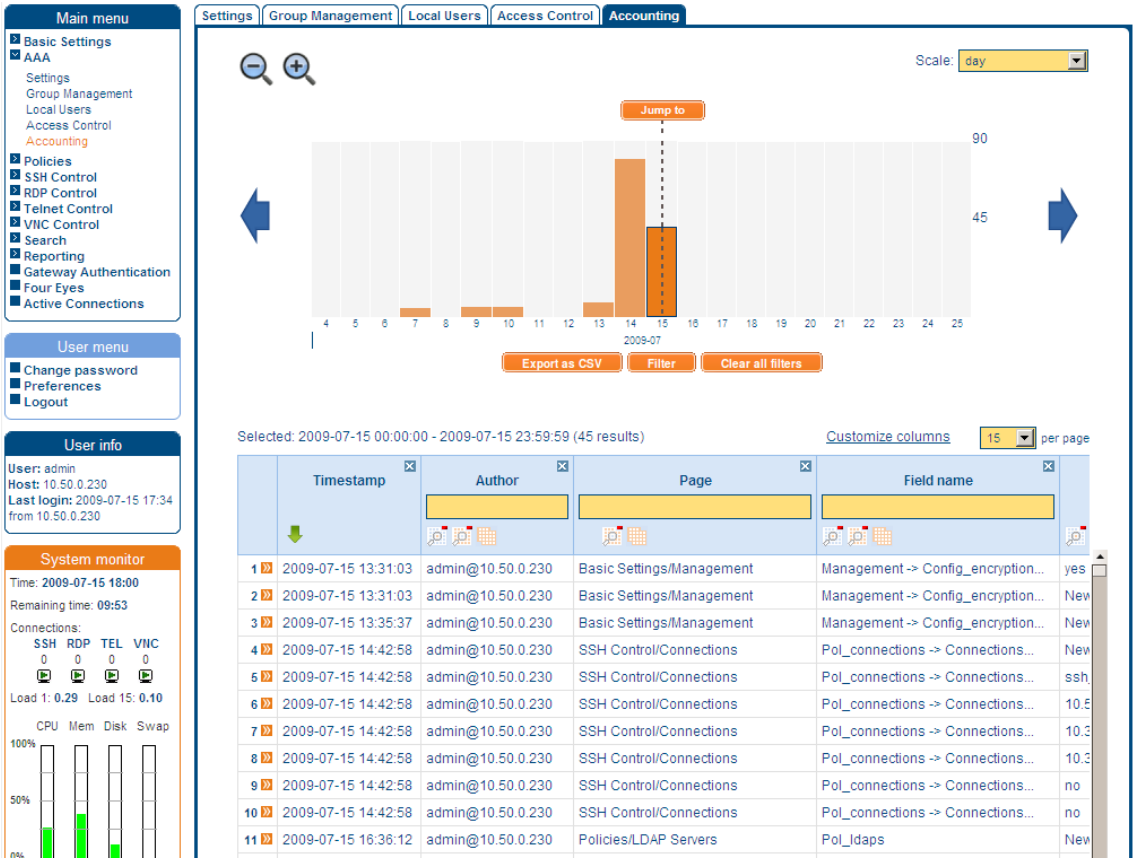


Figure 6.3. Displaying configuration changes

- *Timestamp*: The date when the modification was committed in `YEAR-MONTH-DAY HOUR:MINUTE:SECOND` format.
- *Author*: The SCB user who performed the modification.
- *Page*: The main menu item that was modified (for example `Basic Settings > Management`).
- *Field name*: The name of the field on the page that was modified.
- *New value*: The new value of the field after the modification.
- *Description*: The changelog entered by the SCB administrator. Changelogs are available only if the **AAA > Settings > Require commit log** option was enabled at the time of the change.
- *Old value*: The original value of the field.
- *Swap*: Signs if the order of objects was modified on the page (for example the order of two policies in the list).

6.3. Reports

SCB periodically creates reports on the activity of the administrators, the system-health information of SCB, as well as the processed traffic. These reports are available in Portable Document (PDF) format by selecting **Reporting > Reports** from the Main Menu. The reports are also sent to the e-mail address set at **Basic Settings > Management > Mail settings > Send reports to**, unless specified otherwise in the configuration of the report.

To access the reports from the SCB web interface, the user must have the appropriate privileges.

Note
If the **Basic Settings > Management > Mail settings > Send reports to** address is not set, the report is sent to the SCB administrator's e-mail address.

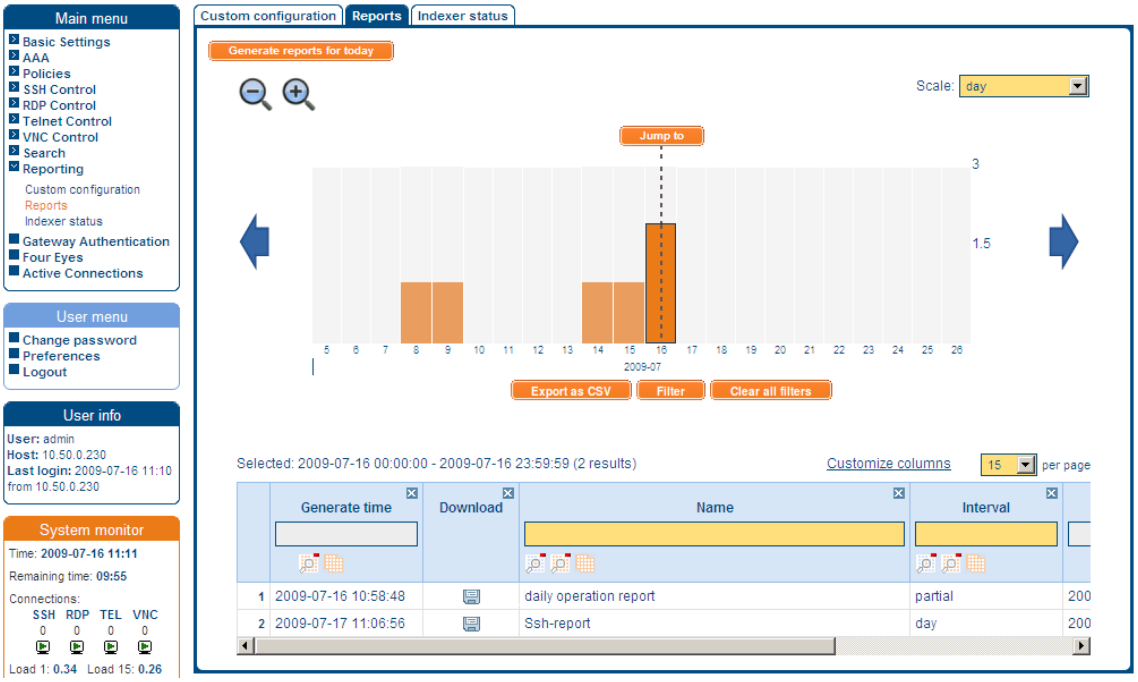


Figure 6.4. Browsing reports

Reports are generated as follows:

- *Daily reports* are generated every day at 00:01.
- *Weekly reports* are generated every week on Monday at 00:01.
- *Monthly reports* are generated on the first day of every month at 00:01.

Tip
Use the time bar to find reports that contain a particular period. If you select a period (for example click on a bar), only those reports will be displayed that contain information about the selected period.

The reports are available in Adobe Portable Document Format (PDF), and contain the following information for the given period:

- *Configuration changes*: Lists the number of SCB configuration changes per page and per user. The frequency of the configuration changes is also displayed on a chart.

- *Main reports*: Contains statistics about the total traffic that passed SCB, including the number of sessions that passed for every connection policy, the used usernames, clients, and servers, and so on.

Note
Connections that are still in progress when the report is generated are excluded from the report.

- *Reports by connection*: Contains separate statistics about every connection policy configured on SCB.
- *System health information*: Displays information about the filesystem and network use of SCB, as well as the average load.

The following information is available about the reports:

- *Download*: A link to download the report.
- *Name*: Name of the report.
- *Interval*: The length of the reported period, for example week, month, and so on.
- *Report from*: The start of the reported interval.
- *Report to*: The end of the reported interval.
- *Generate time*: The date when the report was created.

6.4. The SCB connection database

The connection database detailing the various meta-information about connections and connection-requests is available on the **Search > Search** page. The search queries can include only alphanumerical characters.

Warning
Only users with the required privileges can access the Search page. The following users can access the Search page:

- Members of groups who have the `Search` privilege set.
- Members of groups who have the `Audit` or `Audit&Authorize` permission set in the **Access Control** field of a connection policy. These users can access only the audit trails of the respective connections. Unless modified by the administrator of SCB, members of the `search` group can access every audit trail by default.
- The `admin` user.

Note
Starting with SCB 3.0, data about archived connections can be automatically deleted from the connection database. For details, see *Procedure 5.1.12, Configuring cleanup for the SCB connection database (p. 147)*.

In addition to the regular search interface of SCB, the following features are also available:

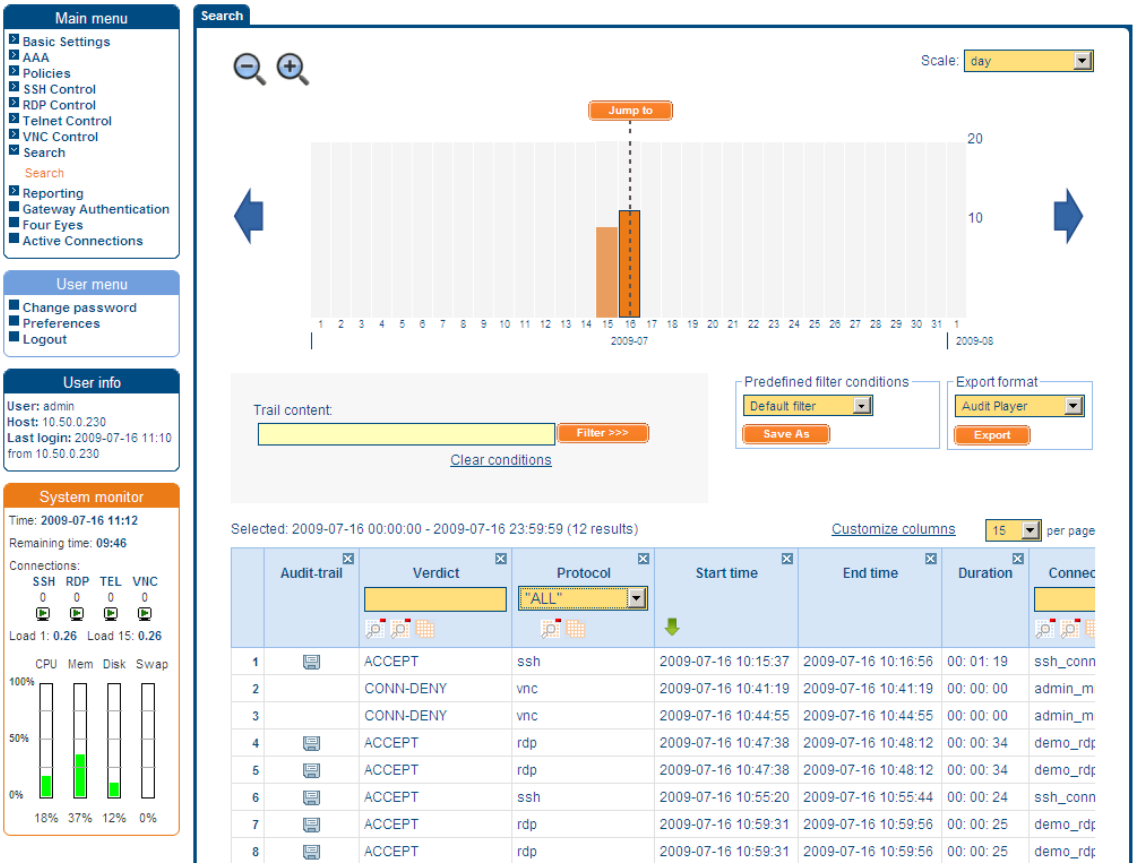


Figure 6.5. Browsing the connections database

- The search results can be exported as a comma-separated text file. Select **Export format > CSV**, and click **Export**.
- To process the search results later with the Audit Player application, the search results can be exported in a special format. Select **Export format > Audit Player**, and click **Export**. When you open this file in the Audit Player application, SCB will download the audit trails corresponding to the search results.
- Create filters. For details, see *Procedure 6.4.2, Creating predefined filters (p. 185)*.
- Search in the content of the indexed audit trails. Enter your search keywords into the **Trail content** field, and click **Filter**. Note that the search is case insensitive. This feature is available only if auditing and full-text indexing was requested for the connection. For details, see *Procedure 6.7, Configuring full-text indexing of audit trails (p. 189)*.

6.4.1. Connection metadata

SCB stores the following parameters about the connections:

- *Archive date*: The date when the connection was archived or cleaned up.
- *Archive path*: The path where the audit trail was archived on the remote server.
- *Archive server*: The hostname or IP address of the remote server where the audit trail was archived.

- *Audit trail*: Name and ID of the audit file storing the traffic of the channel. If the session has an audit trail, a download icon is displayed. Note that a the following letters may appear on the download icon:
  - *C*: The audit trail has been cleaned up and is not available any more.
  - *A*: The audit trail has been archived. SCB will try to retrieve it from the archive server.
  - *X*: The audit trail is not available for some reason.
- *Authentication Method*: The authentication method used in the connection.
- *Channel Policy*: The channel policy applied to the channel opening request.
- *Channel Type*: Type of the channel.
- *Commands*: The list of commands that the user issued in a terminal session. Available only for Telnet and SSH session shell connections, if an Audit Player indexer service has processed the audit trail. For details on configuring Audit Player indexing, see *Procedure 6.7, Configuring full-text indexing of audit trails (p. 189)*.
- *Connection ID*: The identifier of the connection.
- *Connection Policy*: The connection policy applied to the client's connection request.
- *Destination IP*: The IP address of the server as requested by the client.
- *Destination Port*: The port number of the server as requested by the client.
- *Device Name*: The name or ID of the shared device (redirect) used in the RDP connection.
- *Duration*: The length of the session.
- *Dynamic Channel*: The name or ID of the dynamic channel opened in the RDP connection.
- *End Time*: Date when the channel was closed.
- *Environment*: The environment variables sent by the client.
- *Executed Command*: The command executed in a **Session exec** channel.
- *File Operations*: The list of file operations (for example, file upload, create directory) performed by the client. Available only for SCP and SFTP sessions (**Session exec SCP** and **Session SFTP** SSH channels) if the **Log file transfers to database** option is enabled in the Channel Policy of the connection.

Note
For SFTP connections, this field includes the path and the filename. For SCP connections, it includes only the filename; the path is available in the **SCP Path** field.

Windows and UNIX systems use different separator characters in the pathname, backslash (\) and slash (/), respectively. As the SCP and SFTP protocols do not specify the separator character used, SCB uses slash (/), for example, `/var/log/messages`.

Tip
Use the filter in the header of the column to find sessions containing a specific file (for example, enter `.gz` to list sessions that accessed files with the .gz extension). Note that currently it is possible to search only in the filename and path, and not in the changed privileges.

- *Four-eyes Authorizer*: The username of the user who authorized the session. Available only if 4-eyes authorization is required for the channel. For details on 4-eyes authorization, see *Section 8.3, Configuring 4-eyes authorization (p. 215)*.
- *Four-eyes Description*: The description submitted by the authorizer of the session.
- *Port-forward Target IP*: The traffic was forwarded to this IP address in **Remote Forward** and **Local Forward** channels.
- *Port-forward Target Port*: The traffic was forwarded to this port in **Remote Forward** and **Local Forward** channels.
- *Port/X11 forward Originator IP*: The IP address of the host initiating the channel in **Remote Forward** and **Local Forward** channels. Note that this host is not necessarily the client or the server of the SSH connection.
- *Port/X11 forward Originator Port*: The number of the forwarded port in **Remote Forward** and **Local Forward** channels.
- *Protocol*: The protocol used in the connection (SSH, RDP, or Telnet).
- *Rule number*: The number of the line in the channel policy applied to the channel.
- *SCP Path*: Name and path of the file copied via SCP. Available only for SCP sessions (**Session exec SCP** SSH channels) if the **Log file transfers to database** option is enabled in the Channel Policy of the connection.

Note
This field includes only the path; the filename is available in the **File Operations** field.

Windows and UNIX systems use different separator characters in the pathname, backslash (\) and slash (/), respectively. As the SCP and SFTP protocols do not specify the separator character used, SCB uses slash (/), for example, `/var/log/messages`.

- *Server IP*: The IP address of the server connected by SCB.
- *Server-local IP*: The IP address of SCB used in the server-side connection.
- *Server-local Port*: The port number of SCB used in the server-side connection.
- *Server Port*: The port number of the server connected by SCB.
- *Session ID*: ID number of the TCP session.
- *Source IP*: The IP address of the client.
- *Source Port*: The port number of the client.
- *Start Time*: Date when the channel was started.
- *Subsystem Name*: Name of the SSH subsystem used in the channel.
- *Username on server*: The username used to log in to the remote server. This username can differ from the client-side username if usermapping is used in the connection. For details on usermapping, see *Procedure 8.1, Configuring usermapping policies (p. 208)*.
- *Verdict*: Indicates if the channel was accepted (`ACCEPT`), denied `DENY`, or failed (`FAILED`) for some reason. It can also indicate that the connection was rejected (`CONN-DENY`), failed (that is, it was allowed

to pass SCB but timed out on the server — *CONN-FAIL*), the user authentication failed (*CONN-AUTH-FAIL*), or that the public-key authentication failed in SSH (*CONN-KEY-FAIL*).

Note
The **Verdict** column only accepts capital letters.

6.4.2. Procedure – Creating predefined filters

Purpose:

To save the currently selected filter conditions for later use, complete the following steps:

Steps:

Step 1. Navigate to the **Search > Search** page, and set the filters you need.

Step 2. Select **Predefined filter conditions > Save As**. A popup window is displayed.

Step 3. Enter a name for the filter into the **Name** field.

Step 4. To modify the timeframe of the search, select **Interval**, and set the beginning and ending date and time of the search.

Step 5. By default, the filter is available only for the user who created it. To make it available for others, select **Scope > Global**, click ⊞, and enter the name of the group whose members may use the filter. Repeat this step to add other groups if needed.
To make the filter available for every SCB user, do not specify any usergroup, just leave the field empty.

Note
Filters cannot be modified later, only deleted. A filter can be deleted by the user who created it, and by users whose group has the **Search > Manage global filters** privilege.

Step 6. Click **OK**.

6.5. Procedure – Configuring custom reports

Purpose:

SCB can send the audit trails to the Audit Player application (AP) for processing. AP extracts the text from the audit trails and segments it to tokens. A token is a segment of the text that does not contain whitespace: for example words, dates (*2009-03-14*), MAC or IP addresses, and so on. AP then returns the extracted tokens to SCB, where SCB creates a comprehensive index from the tokens of the processed audit trails, and creates reports from the results. The reports include statistics of the occurrences of the search keywords, as well as screenshots from the audit trail.

To configure SCB to create custom reports, complete the following steps.

Warning
Custom reports require a separate server running the Audit Player application in service mode. For details on setting up the service, see *Procedure 7.1, Installing the Audit Player application (p. 191)*.

If the audit trails are encrypted, ensure that the required decryption keys are available on the AP hosts.

Steps:

Note
Only audit trails created after the custom report has been configured will be processed. It is not possible to create reports from already existing audit trails.

Step 1.

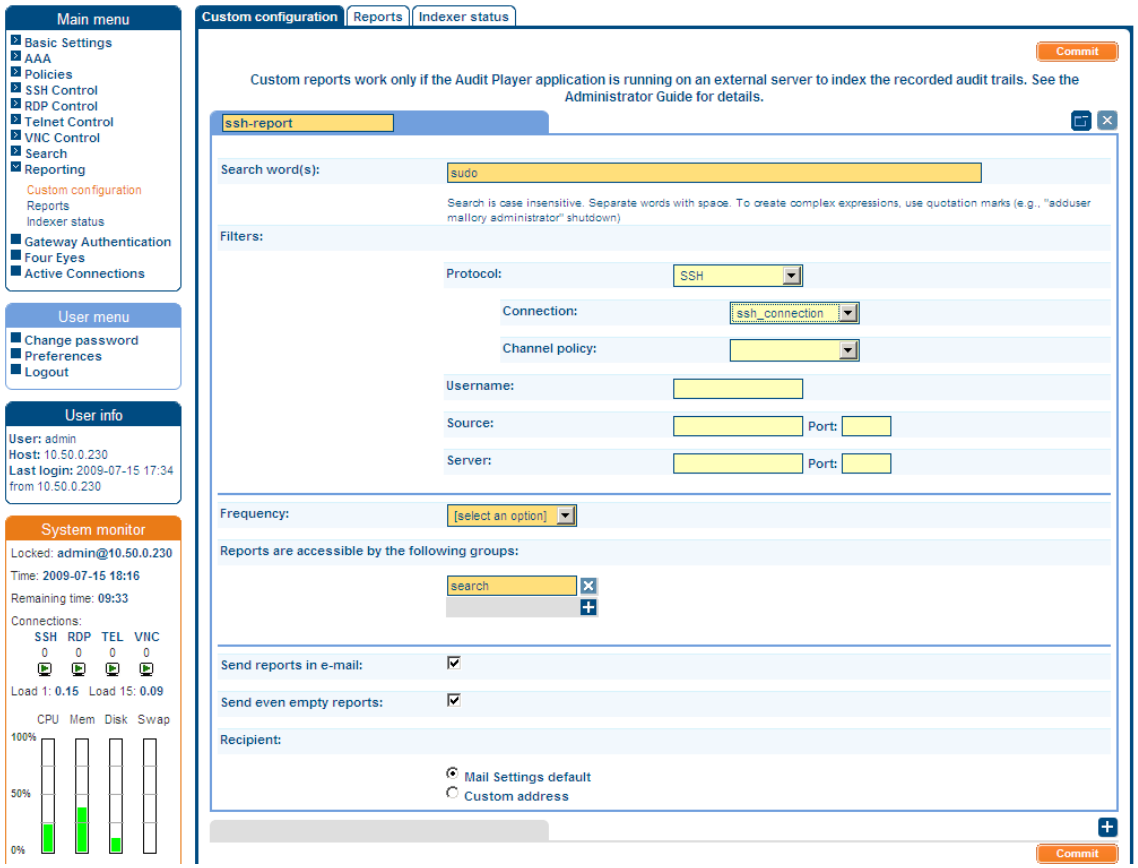


Figure 6.6. Configuring custom reports

Login to the SCB web interface, and navigate to **Reports > Custom configuration**.

Step 2. Click ⊞ and enter a name for the custom report.

Step 3. Enter the search keywords (or parts of the words) into the **Search word(s)** field. Note that the search is not case sensitive. Wildcards and regular expressions are not supported. To search for an exact phrase or expression, enclose the keywords in double quotes, for example `"program files"`.

Note
When using custom reports, AP will extract only the specified search keywords, other texts found in the audit trails will not be indexed. To extract every text from the audit trails, use full-text indexing (for details, see *Procedure 6.7, Configuring full-text indexing of audit trails (p. 189)*). Note that custom reports require much less resources (hard disk, CPU) than full-text indexing.

Step 4. Configure filters to select the audit trails to index. The following filters are available:

- *Protocol*: Process only audit trails of the specified traffic type (for example SSH).
- *Connection*: Process only audit trails of the specified connection policy.
- *Channel policy*: Process only audit trails of the specified channel policy.
- *Username*: Process only audit trails where the specified username was used in the connection. Available only for protocols where the username is known (for example SSH).
- *Source*: Process only audit trails where the specified client IP address or port was used.
- *Server*: Process only audit trails where the specified server IP address or port was used.

Note
If you do not configure any filters, every available audit trail will be processed. Audit trails are created only for channels where the **Audit** option is enabled for the particular channel in the channel policy.

Step 5. Select how often shall SCB create the report from the **Generate this report every** field. Weekly reports are created on Mondays, while monthly reports on the first day of the month. If you want to generate the report only manually, leave this field empty.

Step 6. By default, members of the `search` group can access the custom reports via the SCB web interface. To change this, enter the name of a different group into the **Reports are accessible by the following groups** field, or click ⊞ to grant access to other groups.

Note
Members of the listed groups will be able to access only these custom reports even if their groups does not have read access to the **Reporting > Reports** page. However, only those reports will be listed, to which their group has access to.

Step 7. By default, SCB sends out the reports in e-mail to the address set in the **Basic Settings > Management > Mail settings > Send reports to** field.

Note
If this address is not set, the report is sent to the SCB administrator's e-mail address.

- To disable e-mail sending, unselect the **Send reports in e-mail** option.

- To receive e-mails only when at least one audit trail matching the search criteria was found, unselect the **Send even empty reports** option.
- To e-mail the reports to a different address, select **Recipient > Custom address**, and enter the e-mail address where the reports should be sent.

Step 8. Click **Commit**.

6.6. Monitoring the status of AP indexing services

SCB can cooperate with multiple Audit Player (AP) hosts that process the audit trails. This is needed especially on high-traffic systems, where many audit trails are generated, and also if the audit trails need lot of processing, for example there are many custom reports configured, or full-text indexing is requested for many connections.

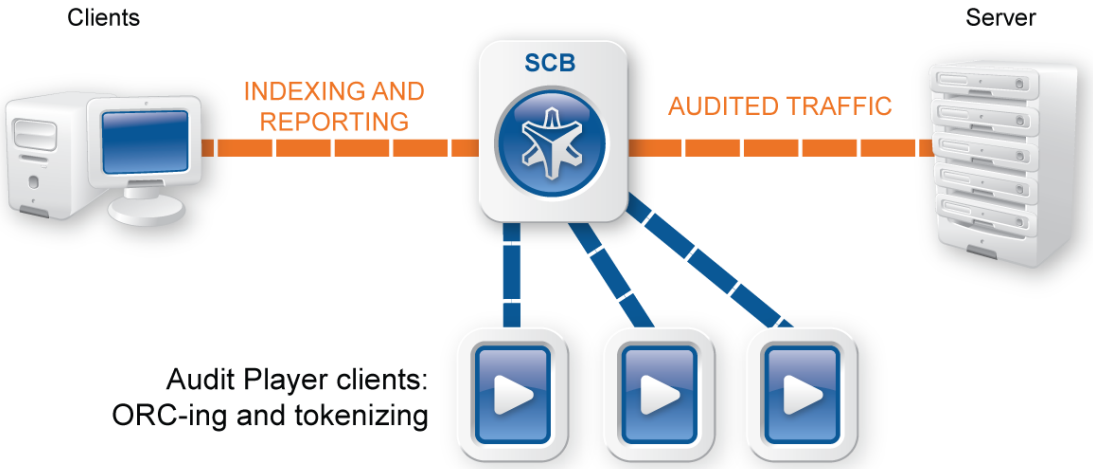


Figure 6.7. Indexing and audit trail processing

The status of the audit-trail processing is displayed on the **Reporting > Indexer status** page. The **Registered Audit Players** section lists the hosts that run the AP application in service mode and are currently connected to SCB. The **Remaining tasks** section lists the audit trails waiting to be processed.

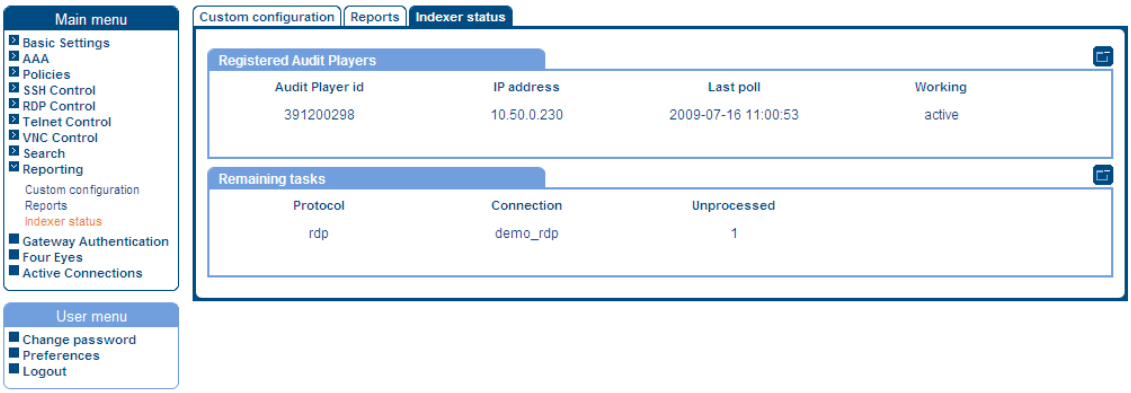


Figure 6.8. Monitoring the status of the Audit Player clients

6.7. Procedure – Configuring full-text indexing of audit trails

Purpose:

SCB can send the audit trails to the Audit Player application (AP) for processing. AP extracts the text from the audit trails and segments it to tokens. A token is a segment of the text that does not contain whitespace: for example words, dates (`2009-03-14`), MAC or IP addresses, and so on. AP then returns the extracted tokens to SCB, where SCB creates a comprehensive index from the tokens of the processed audit trails. That way the contents of the processed audit trails (for example commands typed or texts seen by the user) can be searched from the web interface. To configure full-text indexing, complete the following steps.

Warning
Custom reports require a separate server running the Audit Player application in service mode. For details on setting up the service, see *Procedure 7.1, Installing the Audit Player application (p. 191)*.

Warning
Full-text indexing is a resource intensive (CPU and hard disk) operation both for SCB and the AP hosts, and depending on the number of processed audit trails and parallel connections passing SCB, may affect the performance of SCB. Test it thoroughly before enabling it in a production environment that is under heavy load.

Note
Only audit trails created after the full-text indexing has been configured for the connection policy will be processed. It is not possible to process already existing audit trails.

Steps:

Step 1. Navigate to the **Control** page of the traffic type (for example **SSH Control**), and select the connection policy to index.

Step 2. Select **Enable indexing** and click **Commit**.

Step 3. Check which channel policy is used in the connection, and navigate to the **Connection policies** page.

Step 4. Select the channel policy used in the connection to index, and verify that the **Audit** option is selected for the channels you want to index (for example the `Session shell` channel in SSH, or the `Drawing` channel in RDP).

Step 5. Click **Commit**.

Tip
After that, start a session that uses this connection policy: connect from a client to a server.

When the session is finished, navigate to the **Reporting > Indexer status** page to verify that an AP host is processing the audit trail.

If the audit trails are encrypted, ensure that the required decryption keys are available on the AP hosts.

Chapter 7. Viewing session information and replaying audit trails

SCB records information about the passing sessions into its connection database. Session information can be displayed online from the SCB web interface (for details, see *Chapter 6, Browsing log messages and SCB reports (p. 175)* and *Section 6.4, The SCB connection database (p. 181)*). The Audit Player (AP) is a desktop application that can replay recorded audit trails, much like a media player replays movie files.

AP is available for the following 32-bit and 64-bit platforms:

- Microsoft Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7

The AP application can currently replay the following session types:

- SSH terminal sessions
- Remote X11 sessions forwarded within the SSH traffic. Note that not the entire desktop is displayed, only the windows of the remotely-accessed application.
- The Drawing channel (that is, the desktop) of RDP4, RDP5, RDP6, and RDP7 sessions (except for remote desktops that use the Aero graphical interface)
- Telnet and TN3270 terminal sessions
- VNC sessions
- SCP sessions
- SFTP sessions

For details on how to install the Audit Player application, see *Procedure 7.1, Installing the Audit Player application (p. 191)*.

For details on how to replay audit trails, see *Section 7.2, Replaying audit trails (p. 194)*.

For details on how to search audit trails, see *Section 7.3, Using AP (p. 198)*.

For details on how to use AP to automatically download and process audit trails from SCB, see *Procedure 7.1, Installing the Audit Player application (p. 191)* and *Procedure 6.7, Configuring full-text indexing of audit trails (p. 189)*.

7.1. Procedure – Installing the Audit Player application

Purpose:

To install the Audit Player application, complete the following steps.

Warning
Installing Audit Player requires administrator privileges. The installed Audit Player can be run by restricted users as well if the required privileges are set.

Steps:

Step 1. Download the Audit Player application from *the BalaBit web site*.

Step 2. Start the downloaded file.

Step 3. Read the end-user agreement of AP and click **I Agree**.

Step 4. Some fonts installed by AP have a separate copyright license. Click **I Agree** to accept it.

Step 5. Select the folder where AP will be installed and click **Install**.

Step 6. The installer can automatically install and register several fonts to improve the processing of X11 audit trails. If you will use AP to replay X11 audit trails, you are recommended to install these fonts.

Note
X11 fonts are installed only by Audit Player version 2011.2 and later. Note that installing these fonts automatically opens the `Windows\Fonts` folder in the Windows Explorer. This is normal and needed to avoid having to reboot the host. You can simply close this window.

Step 7. If you will run AP in service mode to automatically download and process the audit trails from SCB, complete the following steps. Otherwise, click **Save settings**

Step a. Select **Enable Audit Indexing Service**.

Step b.

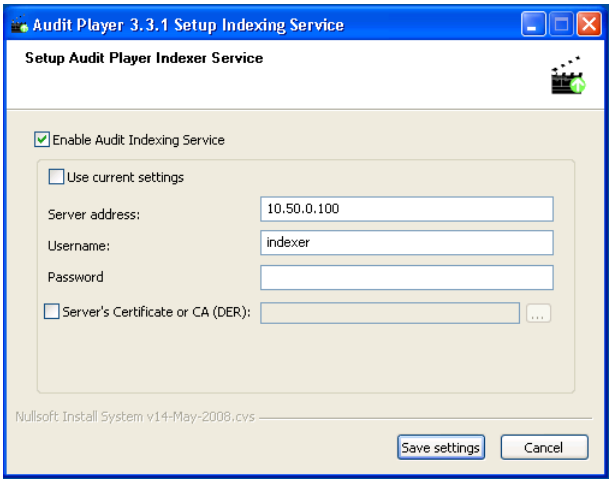


Figure 7.1. Configuring service mode in AP

Enter the IP address or the hostname of your SCB host into the **Server address** field.

Warning
This address must be the same that appears in the Common Name of SCB's certificate. If it is a hostname, it must be resolvable from the host running AP, either via DNS, or locally from a file.

Step c. Enter the **Username** and **Password** of the SCB user account AP will use to access SCB. This user must be a member of the *indexing* group. If LDAP or RADIUS authentication is used on SCB, the user must also exist in the LDAP/RADIUS database.

Step d. To be able to communicate with SCB, install the Server certificate of SCB (the one set in **Basic Settings > Management > SSL certificate > Server Certificate** of the SCB web interface) or the certificate of the certificate authority that signed the certificate of SCB.

Note
The certificate must be in DER format.

Step e. Click **Save settings**.

Step f. When the installation is finished, click **Close**.

Note
To modify the configuration of the Audit Indexer Service, select **Start menu > All Programs > Audit Player > Configure Audit Indexer Service**.

Step g. On the SCB web interface, navigate to **Reporting > Indexer status**. The new Audit Player should appear in the list of **Registered Audit Players**.

Step 8. *Optional step*: If you want to run the Audit Player application without administrator privileges, complete the following steps:

Step a. Allow the nonrestricted user to write and modify the `C:\Program Files\Audit Player\var` folder.

Step b. Allow the nonrestricted user to write and modify the `C:\Program Files\Audit Player\bin\Textract.Log` file. Create the file if it does not exist yet.

Step c. Allow the nonrestricted user to write and modify the `C:\Program Files\Audit Player\bin\Textract.ini` file.

Step 9. *Optional step*: If you want to run the Audit Player on multicore processor, complete the following steps:

Step a. Start `regedit`.

Step b. Navigate to `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\APService`.

Step c. Right click the **ImagePath** key and select **Modify**.

Step d. In the **Value data** field, after the `.exe` file, enter the `-C` switch and the desired value.

Example 7.1. Using the *-C* switch

```
C:\Program Files\Audit Player\bin\audit-indexer-service.exe -C 4
```

Note

To get the optimal performance, set the value to 1-1.5 times of the number of cores.

Step e. Click **OK**.

7.2. Replaying audit trails

To replay an audit trail, the audit trail must be available on the computer running AP, or you must access it on the BalaBit Shell Control Box search interface from a browser on the computer running AP.

- For details on how to download an audit trail from SCB, see *Procedure 7.2.1, Downloading audit trails from SCB (p. 194)*.
- For details on how to replay an audit trail with AP, see *Procedure 7.2.2, Replaying a session with the Audit Player (p. 195)*.

7.2.1. Procedure – Downloading audit trails from SCB

Steps:

Step 1. Login to the SCB web interface.

Step 2. Navigate to the **Search > Search** page.

Step 3. Use the Calendar bar or the **Jump to** option and the filters to locate the session you want to replay.

Step 4. Click the icon in the **Audit trail** column of the session to download the audit trail to your computer. If the **Audit trail** column is not visible, add it to the list of visible columns. For details, see *Procedure 6.1.1, Displaying information about connections (p. 176)*.

If the **Audit trail** column is visible but empty (that is, it does not contain the icon) then no audit trail was recorded from the session, because the channel policy did not have auditing enabled. For details, see *Procedure 5.1.4, Creating and editing channel policies (p. 126)*.

Note

Downloading audit trails requires the **Search > Search in all connections** privilege, or that the group of the user be listed in the **Access Control** section of the connection policy with **Audit** permission.

Tip
To download every audit trail listed in the search results, select the **Export Format > Audit Player**, and click **Export**.

7.2.2. Procedure – Replaying a session with the Audit Player

Purpose:

To replay a session, complete the following steps:

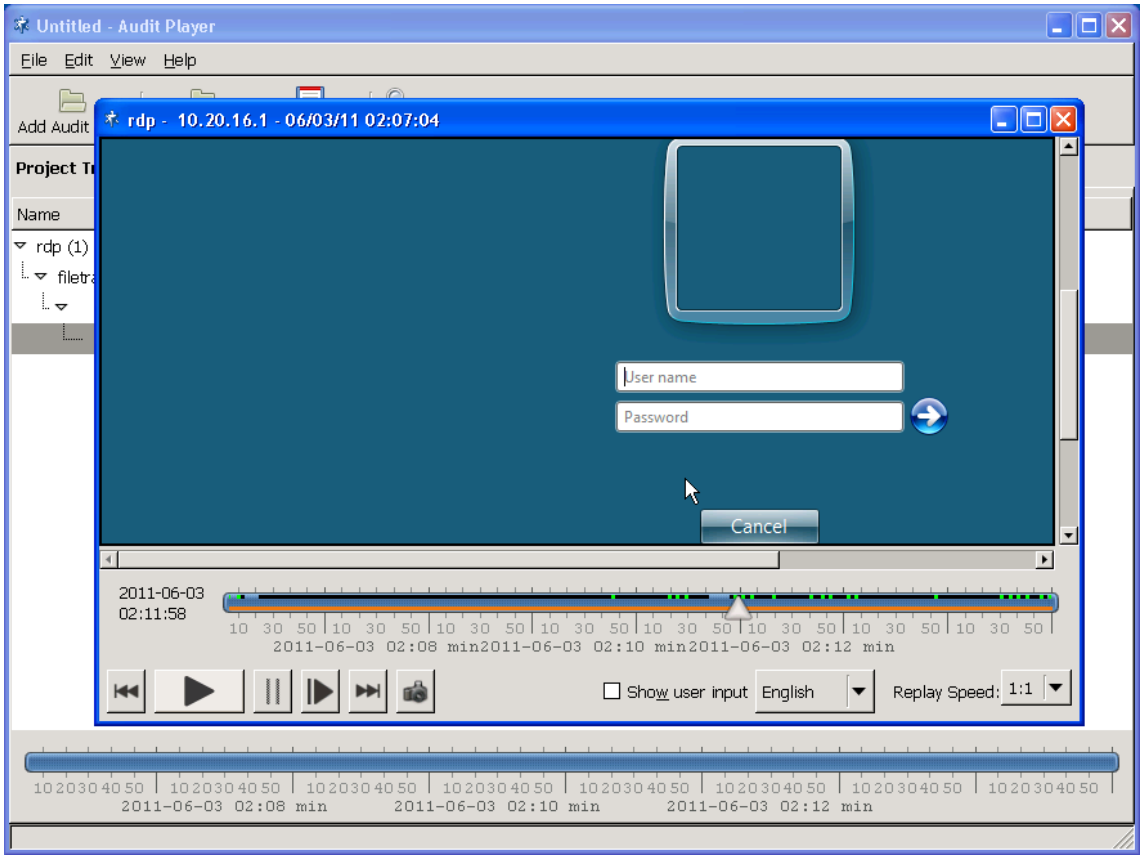


Figure 7.2. Replaying an audit trail

Steps:

Step 1. Open an audit trail to replay using one of the following methods:

- Start the Audit Player application. Select **File > Add Audit Trail** and select an audit trail file.
- Double-click on the audit trail file in Windows Explorer. Note that this method does not work when opening an audit trail that has accented characters in its filename or pathname and the operating system does not natively support accented characters (for example, opening a file called `ellenőrzött-adminisztrátori-kapcsolat.zat` will fail on Windows using English locale).

AP opens the file and displays the sessions stored in the file in the **Project Trails** panel.

Note
To open multiple audit trails, use the Shift and the Control keys.

If the audit trail is encrypted, you must import the private key of the certificate used to encrypt the audit trail before trying to replay the audit trail. For details, see *Section 7.3.3, Replaying and processing encrypted audit trails (p. 202)*.

The extension of the audit trail files is `.log` or `.zat`. Its name consists of `audit-scb-protocolname-timestamp-sequencenumber`.

Step 2. Double-click on the stream you want to replay. The session will be displayed in a new window.

Step 3. Wait until AP processes the stream. The progress of the processing is indicated on the timeline as an orange bar. Click **Play**.

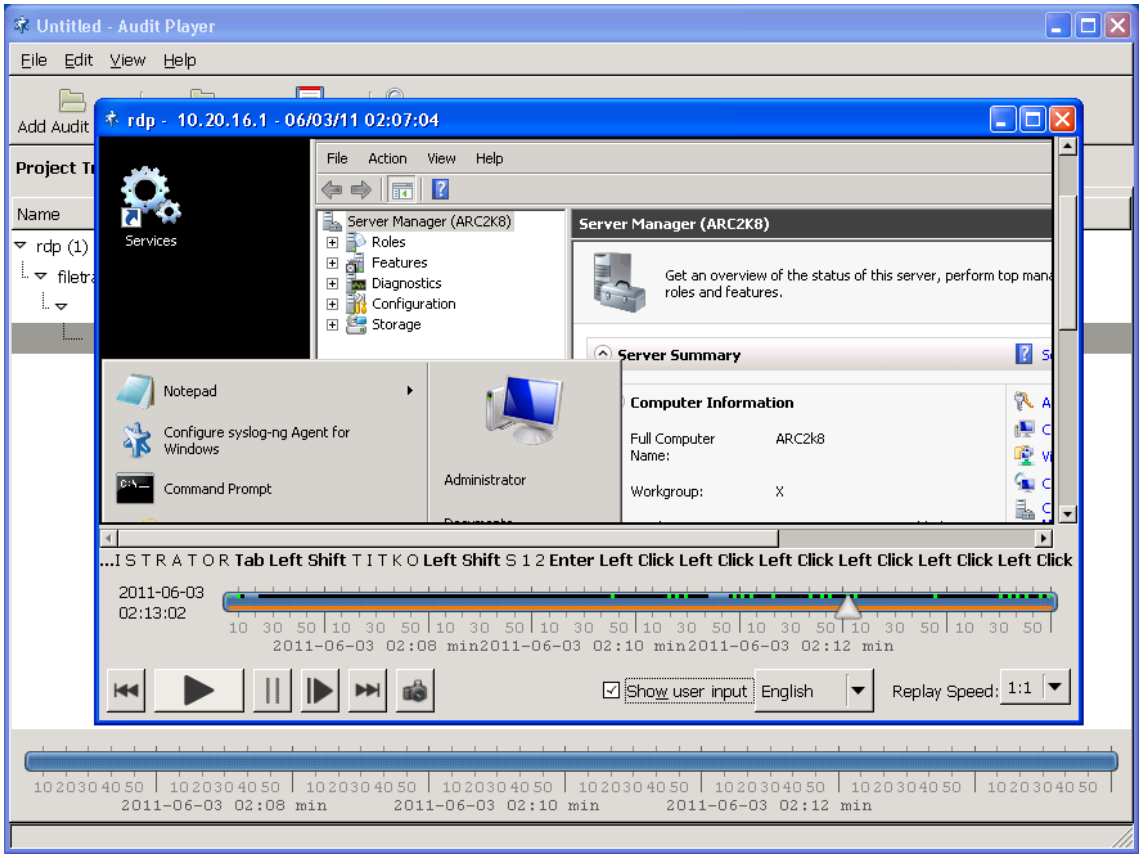


Figure 7.3. Displaying user input

- To adjust the replaying speed, adjust the **Replay Speed** option.
  Starting with Audit Player version 2012.1, the replay speed can be set to `live`. In `live` mode, AP replays the audit trail fast, skipping idle periods, and reverts to real-time replay when reaching the end of the current trail. This is the default replay mode for replaying live streams (clicking **Follow** in the SCB web interface).

- To display the characters (for example commands, passwords, and so on) that the user typed in the session, select **Show user input**. The user input will be displayed above the time bar. Note that the appropriate decryption keys are needed to display the user input if the upstream traffic is encrypted with a different set of certificates.
- To jump to the next event block (where something happens in the protocol), click **Jump to the next event block**. Event blocks are marked with black on the timeline. Two events are considered belonging to the same block, if the distance between them is equal or lower than the 5% of the trail's length. The user input events are marked with green on the timeline.

Tip
The following hotkeys are available during playback:

- *Page Up* / *Page Down*: Jump forward/backward one fifth of the stream.
- *Home* / *End*: Jump to the beginning/end of the stream.
- *Left* / *Right*: Jump forward/backward one tenth of the stream.
- *Up* / *Down*: Jump to the previous/next audit trail.

Tip
To export a session into packet capture (PCAP) format, select the session, then select **Export to PCAP** from the local menu.

7.2.3. Procedure – Replaying SCP and SFTP sessions

Purpose:

To replay SCP and SFTP sessions, complete the following steps:

Steps:

Step 1. Select **File > Add Audit Trail** and select an audit trail file. AP opens the file and displays the sessions stored in the file in the **Project Trails** panel.

Note
SCP and SFTP sessions cannot be searched with **Find**.

To filter for filenames, in the SCB Web interface navigate to **Search > Search**, click **Customize columns...** and add **File operations** to the visible columns. You can then filter for filenames in the filter field of the **File operations** column.

Step 2. Double-click on the session you want to replay. The session will be displayed in a new window.
The session enlists all file transfer and file management actions (for example file deletion, file uploading, changing attributes and so on) on the remote host.

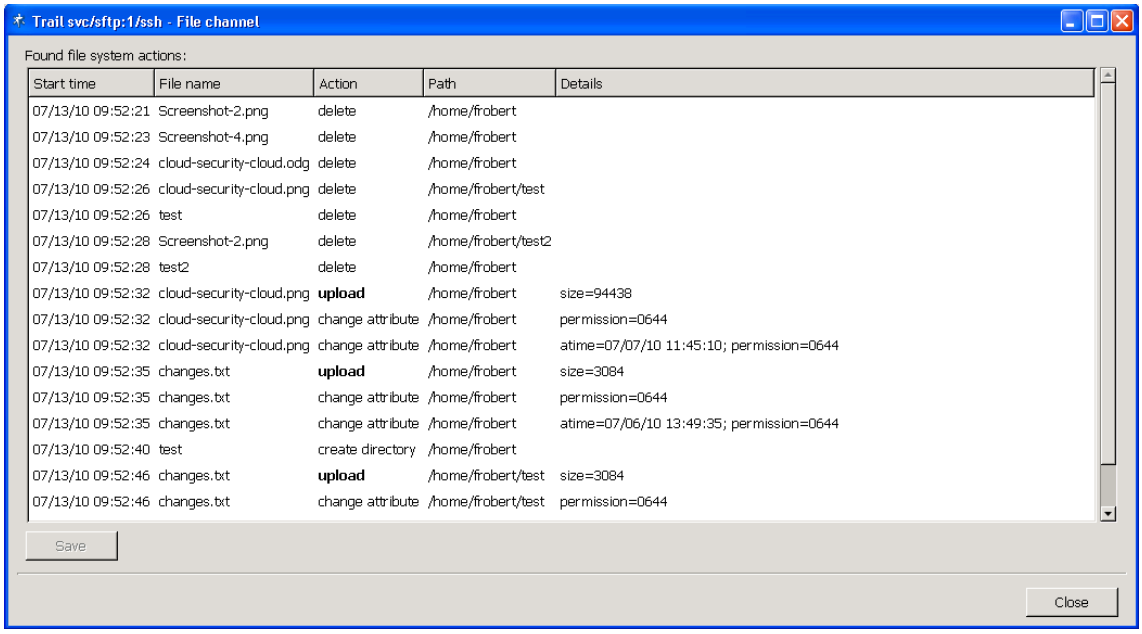

Trail svc/sftp:1/ssh - File channel
Found file system actions:
Start time File name Action Path Details
07/13/10 09:52:21 Screenshot-2.png delete /home/frobert
07/13/10 09:52:23 Screenshot-4.png delete /home/frobert
07/13/10 09:52:24 cloud-security-cloud.odg delete /home/frobert
07/13/10 09:52:26 cloud-security-cloud.png delete /home/frobert/test
07/13/10 09:52:26 test delete /home/frobert
07/13/10 09:52:28 Screenshot-2.png delete /home/frobert/test2
07/13/10 09:52:28 test2 delete /home/frobert
07/13/10 09:52:32 cloud-security-cloud.png upload /home/frobert size=94438
07/13/10 09:52:32 cloud-security-cloud.png change attribute /home/frobert permission=0644
07/13/10 09:52:32 cloud-security-cloud.png change attribute /home/frobert atime=07/07/10 11:45:10; permission=0644
07/13/10 09:52:35 changes.txt upload /home/frobert size=3084
07/13/10 09:52:35 changes.txt change attribute /home/frobert permission=0644
07/13/10 09:52:35 changes.txt change attribute /home/frobert atime=07/06/10 13:49:35; permission=0644
07/13/10 09:52:40 test create directory /home/frobert
07/13/10 09:52:46 changes.txt upload /home/frobert/test size=3084
07/13/10 09:52:46 changes.txt change attribute /home/frobert/test permission=0644
Save
Close

Figure 7.4. SFTP session

Step 3. You can download the uploaded files from the session. These files are marked in the **Action** column with a bold **upload** or **download** label. To download a file, select it and click **Save**.

7.3. Using AP

The main window allows you to sort and organize the audit trails and sessions.

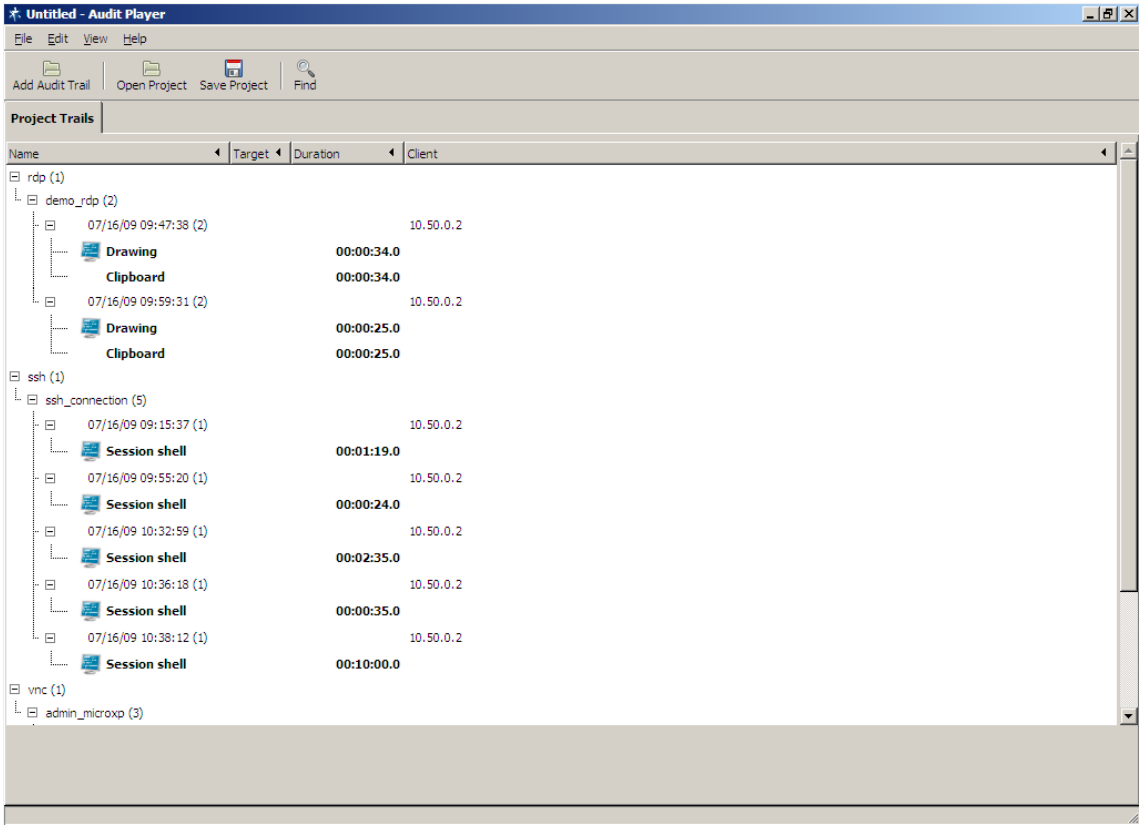


Figure 7.5. The main window of Audit Player

The audit trails loaded into AP are all visible in the **Project Trails** tab. Audit trails are organized into a tree. The tree has the following levels:

- *Protocol*: The type of the audited protocol: SSH, RDP, X11, VNC, SCP, SFTP or Telnet.
- *Connection*: Name of the connection policy that generated the audit trail.
- *Session*: Details of the connection stored in the audit trail, including the date, duration, source, and target of the connection. (Every audit trail contains only a single session.)

Tip
To sort the list of streams, click the header of the columns.

- *Stream*: The traffic that can be replayed. A session may contain several streams (for example an SSH connection may include a terminal-session stream and an X-forward stream).

Tip
Select **View > Show Details** to display additional information about the selected stream in a separate **Stream Details** window. These details include the parameters available in the SCB **Search** page (for details, see *Section 6.4, The SCB connection database (p. 181)*) and other parameters like the size of the desktop or the terminal.

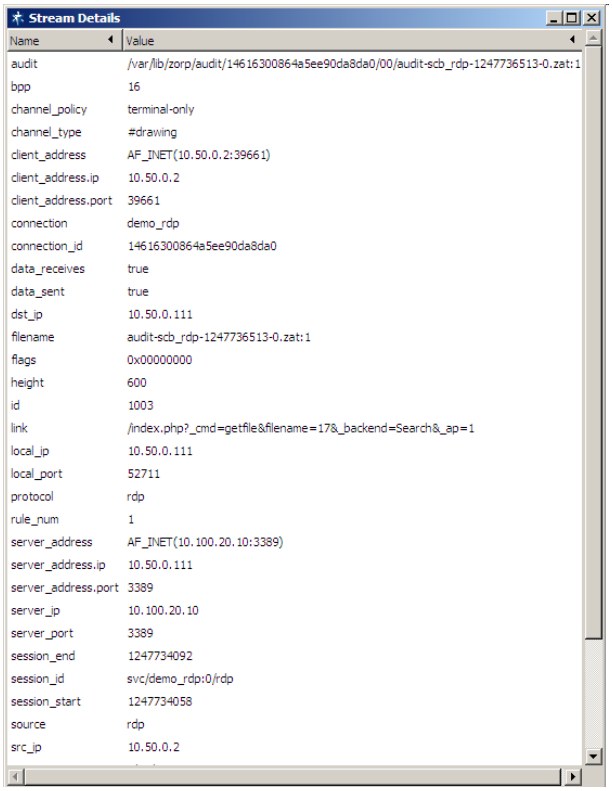


Figure 7.6. The AP Stream Details window

7.3.1. Finding specific audit trails

Organizing the audit trails is simple. Every loaded audit trail is displayed in the **Project Trails** tab. To open a new tab, select **File > New Trailset** and enter a name for the trailset. After that, you can drag-and-drop the interesting streams to this tab, or right-click a stream and select **Copy to another set** or **Move to another set**.

To remove a stream from a trailset, click on the stream and select **Delete** from the local menu. Deleting a stream from the Project Trails tab deletes the stream from every trailset.

To filter the audit trails available in a trailset, select **Edit > Find**, and enter your search keywords into the **Text** field. To search for special keys or events, use the **Key sequence** field.

To search on a broader event scale, select **Any Input Event**, **Any Keyboard Event**, or **Any Mouse Event**, which will mark all events of the selected type. To search for specific keys or mouse clicks (for example **Escape**, or **Double Click**), select the key or mouse click type from the list. Click **Find**. The AP searches for the selected events in the trail and then visualizes them on the timeline with white. To display the time when the event occurred in the trail, hover the mouse over the timeline. The absence of white lines on the timeline means that the selected events were not found in that specific period in the trail.

Warning
When searching audit trails of SSH and Telnet terminal sessions, the character encoding set in **Edit > Preferences > Terminal encoding** can affect the search results: if the session uses a different encoding, special characters might not be recognized and thus will be omitted from the search results.

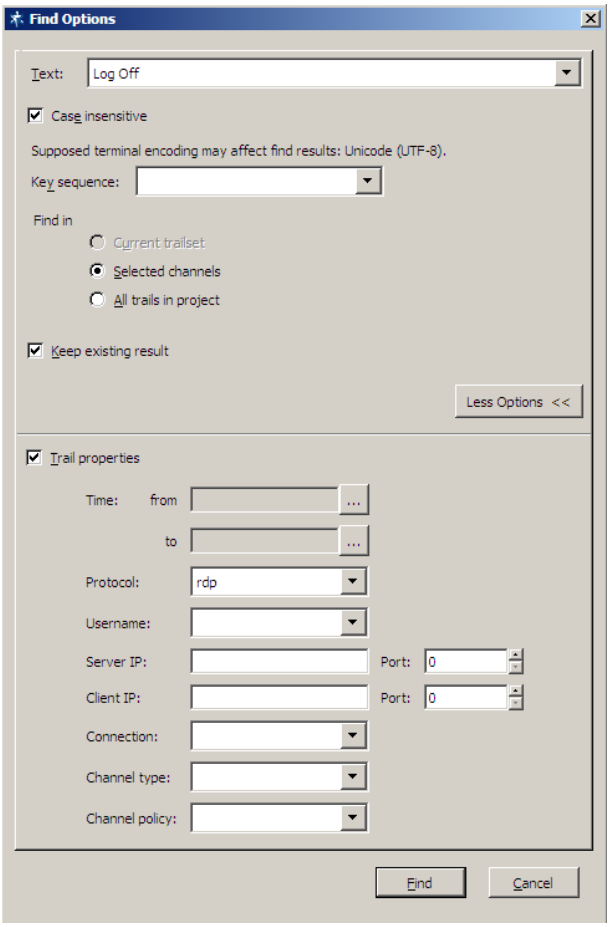


Figure 7.7. Searching audit trails with AP

To search in the metadata of the trails, select **More Options > Trail properties**. The following metadata is available for filtering:

- *Time*: Use the **From** and **To** fields to filter on the duration of the streams matching the other search criteria.
- *Protocol*: The protocol used in the stream: SSH, RDP or Telnet.
- *Username*: The username used in the session (if available).
- *Client IP*: The IP address of the client computer.
- *Client Port*: The port of the client computer used to establish the connection. *0* matches for every port.
- *Server IP*: The IP address of the server connected by SCB.
- *Server Port*: The port of the server connected by SCB. *0* matches for every port.
- *Channel Policy*: The channel policy applied to the session.

- *Connection*: The `session_id` identifying the particular session.
- *Channel Type*: The type of channel used in the stream (for example terminal session, drawing, and so on). See the list of supported channel types in the following sections: *Section 5.2, SSH-specific settings (p. 148)*, *Section 5.3, RDP-specific settings (p. 165)*, and *Section 5.4, Telnet-specific settings (p. 172)*.

7.3.2. Using projects

To save a set of audit files, including all trailsets and search results, select **File > Save Project**, enter a name for the project, and enter a password. The password is used to encrypt the workspace file as it may contain sensitive data.

Warning
AP does not encrypt the audit trail files, only the workspace file. If the audit trails were not encrypted by SCB, sensitive information may be recovered from them.

7.3.3. Replaying and processing encrypted audit trails

To replay encrypted audit trails, the private key of the certificate used to encrypt the audit trails must be available on the machine running AP. This key must be either imported into the Personal Certificate Store / My Store on Windows, or it must be available on a USB token.

To import a private key, select **Edit > Import key**, select the file containing the key, and click **OK**. Then enter the password for the key if needed.

Note
The private key must be in PKCS12 format.

7.3.3.1. Procedure – Importing certificates with MMC

Purpose:

To process encrypted audit trails with the Audit Player application running in service mode, the appropriate private key must be available in the Private certificate store of the Audit Indexer service. To add a certificate to this store, complete the following steps.

Prerequisites:

The certificate and private key to import in PKCS12 format. The `key_usage` parameter of the certificate must be `AT_KEYEXCHANGE`. Certificates with the `AT_SIGNATURE` value will not work.

Tip
To convert a certificate from a different format (for example, `.pfx`) to PKCS12, you can use the OpenSSL software package or a web browser like Internet Explorer or Mozilla Firefox. For details, see *Procedure 7.3.3.2, Converting certificates using OpenSSL (p. 203)* and *Procedure 7.3.3.3, Converting certificates using Firefox (p. 204)*, respectively.

Steps:

Step 1. Start Microsoft Management Console by executing `mmc.exe` (**Start** menu **Run application**).

Note
Running `mmc.exe` requires administrator privileges.

Step 2. Click on the **Add/Remove snap-in** item of the **File** menu.

Step 3. Click **Add**, select the **Certificates** module, and click **Add**.

Step 4. Select **System account** in the displayed window and click **Next**.

Step 5. Select **Audit Indexer** and click **Next**.

Step 6. Select **Local computer** and click **Close**.

Step 7. To import a private key, navigate to **Console Root > Certificates > Audit Indexer > Personal**.

Step 8. Right-click on the **Certificates** folder and from the appearing menu select **All tasks > Import**. The **Certificate Import Wizard** will be displayed. Click **Next**.
Optional step: Certificates used to decrypt the audit trails include a private key. Provide the password for the private key when requested.

Step 9. Click **Finish** on the summary window and **Yes** on the window that marks the successful importing of the certificate.

7.3.3.2. Procedure – Converting certificates using OpenSSL

Purpose:

To convert a certificate and its private key into PKCS12 format (for example, from `.pfx`) using the OpenSSL software package, complete the following steps.

Steps:

Step 1. Convert the `.pfx` file into PEM format. Issue the following command in a terminal console: `openssl pkcs12 -in <certificate>.pfx -out <certificate>.pem -nodes`

Step 2. Open the `<certificate>.pem` file in a text editor, and copy the certificate and the private key into two separate files, for example, `<mycertificate>.pem` and `<mycertificate>.key`.

Step 3. Convert the certificate and the private key into PKCS12 format. Issue the following command: `openssl pkcs12 -export -out <mycertificate>.p12 -inkey <mycertificate>.key -in <mycertificate>.pem`

Step 4. If the original `.pfx` certificate was imported into a certificate store, remove it and import the new certificate file (that is, `<mycertificate>.p12`). Importing certificates is described in *Procedure 7.3.3.1, Importing certificates with MMC (p. 202)*.

7.3.3.3. Procedure – Converting certificates using Firefox

Purpose:

To convert a certificate and its private key into PKCS12 format (for example, from `.pfx`) using the Mozilla Firefox web browser, complete the following steps.

Steps:

Step 1. Start Firefox and select **Edit > Preferences > Advanced > Encryption > View Certificates > Your Certificates > Import**, and import the `.pfx`. If needed, enter the password of the private key as well.

Step 2. Select the newly imported certificate, and click **Backup**.

Step 3. Select the **PKCS12** format and enter a name for the file (for example, `mycertificate`).

Step 4. If the original `.pfx` certificate was imported into a certificate store, remove it and import the new certificate file (that is, `<mycertificate>.p12`). Importing certificates is described in *Procedure 7.3.3.1, Importing certificates with MMC (p. 202)*.

7.3.4. Searching in graphical streams

The Remote Desktop Protocol (RDP), Virtual Network Computing (VNC), and X11 protocols transfer most texts displayed by the remote applications as graphical data. To make these texts searchable, AP automatically processes the opened audit trails and performs optical character recognition (OCR) on the streams.

During installation, AP creates a database from the fonts installed on the system, and uses these fonts for the OCR process. This means that the fonts used by the servers and applications accessed using RDP must be installed on the host running AP. Otherwise AP might not correctly recognize every font type, and miss parts of the text.

Warning
Fonts installed after AP has been installed are not automatically used in the OCR process. To include newly installed fonts, complete *Procedure 7.3.5, Adding a new font to the OCR database (p. 204)*.

Fonts for non-Latin alphabets and symbols, such as Japanese, Chinese, or Webdings can slow down the OCR process, so by default they are excluded from the font database even if these fonts are installed on the host. To enable OCR-ing such fonts, complete *Procedure 7.3.5, Adding a new font to the OCR database (p. 204)*.

Note
The OCR feature of the Audit Player application currently supports only RDP sessions. Certain accented and other special (non-ASCII) characters might not be recognized correctly.

OCR-ing VNC sessions is partially supported, provided that the VNC server is running on Windows.

7.3.5. Procedure – Adding a new font to the OCR database

Purpose:

To add a new font to the AP database, complete the following steps.

Steps:

Step 1. If not already installed, install the font on the host running AP.

Step 2. Fonts for non-Latin alphabets and symbols, such as Japanese, Chinese, or Webdings can slow down the OCR process, so by default they are excluded from the font database even if these fonts are installed on the host. The following fonts are excluded by default: *Algerian**, *Arial Unicode MS*, *Brush Script**, *CommonBullets*, *GulimChe*, *Gulim*, *Marlett**, *Matura MT Script**, *Monotype Sorts*, *MS Gothic*, *MS Hei*, *MS Mincho*, *MS Song*, *MT Extra**, *Playbill**, *Symbol**, *Webdings**, *Wingdings**.
If the font you want to use appears on the list, complete the following steps.

Step a. Open the following file in a text editor (for example, Notepad): `<installation-directory-of-Audit Player>\bin\textract.ini`

Step b. Find the *[Recognition]* section in the file.

Step c. Find keys *Exclude* and *Include2*.

Step d. Remove the fonts you want to use from the *Exclude* list.

Step e. Add the fonts you want to use to the *Include2* list. The elements of the list are separated by comma. The * wildcard character can be used to refer to font families.
For example, to OCR Japanese fonts, remove the *MS Gothic* from the *Exclude* list and add *MS Gothic,MS Gothic*,MS Mincho,MS Mincho*,MS PMincho*,MS PGothic*,MS UI*,Franklin*,Meiryo** to the *Include2* list.

Step 3. Start the Audit Player application and select **Edit > Rebuild OCR database**.
Expected result:

The Audit Player application updates the font database. The newly added fonts will be recognized in audit trails indexed after the database update.

7.4. Troubleshooting the Audit Player

This section provides help on solving common problems of the Audit Player application (AP).

7.4.1. Logging with the Audit Player

By default, the Audit Player application sends error messages to the Application eventlog container. To create more detailed logs that are sometimes needed to troubleshoot a problem, command-line parameters must be specified.

7.4.1.1. Procedure – AP logging in user mode

Purpose:

To create debug logs of AP running in user mode, complete the following steps.

Steps:

Step 1. Select **Start > Run > cmd** to open a command prompt.

Step 2. Navigate to the folder where the Audit Player application is installed, for example `cd "C:\Program Files\Audit Player\bin\"`.

Step 3. Enter the following command: `audit-player.exe --verbose 3 --log-file "audit-player.log"`. The Audit Player application will start and create the specified log file.

Step 4. Reproduce the error in AP and check the created log file.

Step 5. If needed, repeat Step 3 with a higher log level (for example with the `--verbose 7` parameter).

7.4.1.2. Procedure – AP logging as an indexer service

Purpose:

To create debug logs of AP running as an indexer service, complete the following steps.

Steps:

Step 1. Start the service management interface of Windows, for example select **Start > Run > mmc**, then select **File > Add/Remove Snap-in > Add > Services**.

Step 2. Double-click on the **Audit Indexer Service** and select **Stop**.

Step 3. Enter the following parameters into the **Start parameters** field: `-d 3 -f "C:\Program Files\Audit Player\service.log" -D 3 -F "C:\Program Files\Audit Player\audit-player.log"`.

Step 4. Select **Start**. The Audit Indexer Service will start and create the specified log file. Note that the ID of the Indexer Service may be added to the name of the log file as a suffix (*`audit-player.log.001`*).

Step 5. Reproduce the error and check the created log file.

Step 6. If needed, repeat Steps 2-4 with a higher log level (for example with the `-d 7` parameter).

Step 7. If you have finished the troubleshooting, restart the Audit Indexer Service without any parameters to lower the logging level to its default.

7.4.2. Keys and certificates

The Audit Player application requires encryption keys and certificates to use its various features. The following summarizes where the keys and certificates should be stored on Windows for AP to function properly.

When run by a user as a regular application, AP may need the following:

- To connect to SCB and download audit trails, the CA certificate of SCB must be available under **Local Computer > Trusted Root Certification Authorities**. This CA certificate is also used to validate timestamped audit trails.
- To replay encrypted audit trails, the private key of the encrypting certificates must be available under **Current User > Personal Certificate Store**.
- To validate digitally-signed audit trails, the certificates used to sign the audit trail or the respective CA certificates must be available under **Local Computer > Trusted Root Certification Authorities**.

When running as an indexer service, AP may need the following:

- To connect to SCB and download audit trails, the CA certificate of SCB must be available under **Local Computer > Trusted Root Certification Authorities**. This CA certificate is also used to validate timestamped audit trails.
- To replay encrypted audit trails, the private key of the encrypting certificates must be available under **Service (Audit Indexer Service) > APService > Personal Certificate Store**.
- To validate digitally-signed audit trails, the certificates used to sign the audit trail or the respective CA certificates must be available under **Local Computer > Trusted Root Certification Authorities**.

7.4.3. Keyframe building errors

When loading audit trails containing a very large number of objects (typically, audit trails of X11 traffic), the Audit Player application might stop processing the audit trail and terminate. This is caused by reaching the GDI Object Handle Limit, which is 10000 by default.

To overcome this problem, increase the value of the *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\GDIProcessHandleQuota* registry key (up to the maximum of 65535) and reboot the computer.

Chapter 8. Advanced authentication and authorization techniques

This chapter describes the advanced authentication and authorization techniques available in SCB.

- For details on creating usermapping policies, see *Procedure 8.1, Configuring usermapping policies (p. 208)*.
- For details on configuring gateway authentication, see *Section 8.2, Configuring gateway authentication (p. 210)*.
- For details on configuring 4-eyes authorization and real-time monitoring, see *Section 8.3, Configuring 4-eyes authorization (p. 215)*.

8.1. Procedure – Configuring usermapping policies

Purpose:

For SSH and RDP connections, usermapping policies can be defined. A usermapping policy describes who can use a specific username to access the remote server: only members of the specified local or LDAP usergroups (for example `administrators`) can use the specified username (for example `root`) on the server. To configure usermapping, complete the following steps.

Warning
In SSH connections, the users must use the following as their username: `gu=username@remoteusername`, where `username` is the username used in the LDAP directory, SCB will use this username to determine their group memberships, and `remoteusername` is the username they will use on the remote server. For example, to access the example.com server as root, use:

```
gu=yourldapusername@root@example.com
```

Note
For RDP connections, usermapping is possible only when gateway authentication is used as well.

Steps:

Step 1.

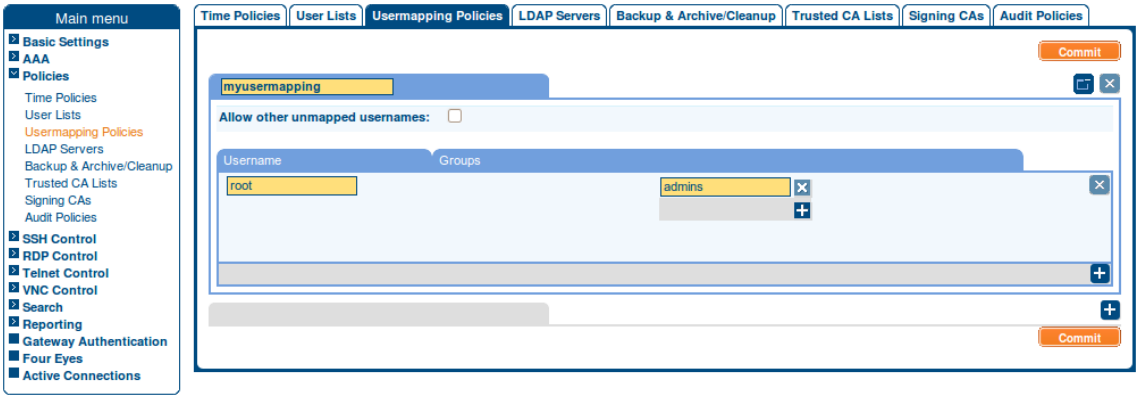


Figure 8.1. Configuring usermapping policies

Navigate to **Policies > Usermapping Policies**.

Step 2. Click ⊞ to create a new policy, and enter a name for the policy.

Step 3. Click ⊞ and enter the username that can be used to access the remote server (for example `root`) into the **Username** field. SCB will use this username in the server-side connection. To permit any username on the server side, enter an asterisk (*).

Step 4. Select **Groups >** ⊞, and specify who is permitted to use the remote username set in the **Username** field.

- If you have an LDAP Server set in the connection policy where you will use usermapping, enter the name of the local or LDAP usergroup (for example `admins`) whose members will be permitted to use the remote username. For details on LDAP authentication, see *Procedure 5.1.7, Authenticating users to an LDAP server (p. 131)*.

Note
The LDAP server configured in the connection policy is not necessarily the same as the LDAP server used to authenticate the users accessing the SCB web interface.

- If you do not authenticate the connections to an LDAP server, enter the name of the userlist whose members will be permitted to use the remote username. For details on using userlists, see *Procedure 5.1.6, Creating and editing user lists (p. 130)*.

Repeat this step to add further groups if needed.

Step 5. Repeat steps 3-4 to add further usernames if needed.

Step 6. Click **Commit**.

Step 7. Navigate to the **Connections** page of the traffic (for example to **SSH Control > Connections**), and select the connection policy to modify.

Step 8. Select the usermapping policy created in Step 2 from the **Usermapping policy** field.

Step 9. Click **Commit**.

Note
For RDP connections, usermapping is possible only when gateway authentication is used as well. When configuring usermapping for RDP connections, proceed to *Procedure 8.2.1, Configuring outband gateway authentication (p. 210)* and configure gateway authentication.

8.2. Configuring gateway authentication

When gateway authentication is required for a connection, the user must authenticate on SCB as well. This additional authentication can be performed out-of-band on the SCB web interface for every protocol. For details about the concepts of gateway authentication, see *Procedure 2.7, The gateway authentication process (p. 16)*.

Note
For SSH connections, gateway authentication can be performed also inband, without having to access the SCB web interface. Inband gateway authentication must be performed when client-side authentication is configured. For details on configuring client-side authentication, see *Section 5.2.3.2, Client-side authentication settings (p. 156)*.

Note
Gateway authentication can be used together with other advanced authentication and authorization techniques like 4-eyes authorization, client- and server-side authentication, and so on.

For SSH connections, the gateway authentication can be performed inband within the protocol, without having to access the SCB web interface.

Warning
If the username used within the procotol is different from the username used to access the SCB web interface to perform gateway authentication, usermapping must be configured for the connection. For details on usermapping, see *Procedure 8.1, Configuring usermapping policies (p. 208)*.

8.2.1. Procedure – Configuring outband gateway authentication

Purpose:

To configure gateway authentication, complete the following steps.

Warning

- The `admin` user is special SCB user and not a member of any user groups; nor can it belong to any group. Since usermapping policies are based on user groups, performing gateway authentication with the `admin` user is likely to result in usermapping errors.
- When using SSL-encrypted RDP connections, or connections that use the Credential Security Service Provider (CredSSP) authentication method, some Microsoft RDP clients restart the connection during the authentication process. This would require the user to perform gateway authentication on the SCB web interface twice. To avoid this situation, SCB temporarily caches the successful gateway authentication results if the client terminates the connection at a certain step while establishing the connection. The cache is used to automatically authenticate the restarted connection without user interaction.
  In this case, restarted connection coming from the same source IP and targeting the same destination IP:port pair will be authenticated from the cache. The cache is deleted after three minutes, or when a connection is authenticated from the cache.

  However, caching the authentication results has the following side-effect: if a different connection targets the same destination IP:port pair from seemingly the same source IP address within the brief period when SCB

expects the original connection to be re-established, the new connection can access the target server without having to authenticate on the SCB gateway. Normally, this can occur only if the clients are behind a NAT.

- If the clients are behind a device that performs network address translation (NAT), SCB will see as if every connection was initiated from the same IP address. Therefore, in such cases using outband gateway authentication is not recommended for security reasons, especially for RDP connections. If possible, use inband gateway authentication instead.

Steps:

Step 1. Navigate to the **Connections** page of the traffic (for example to **SSH Control > Connections**), and select the connection policy to modify.

Step 2.

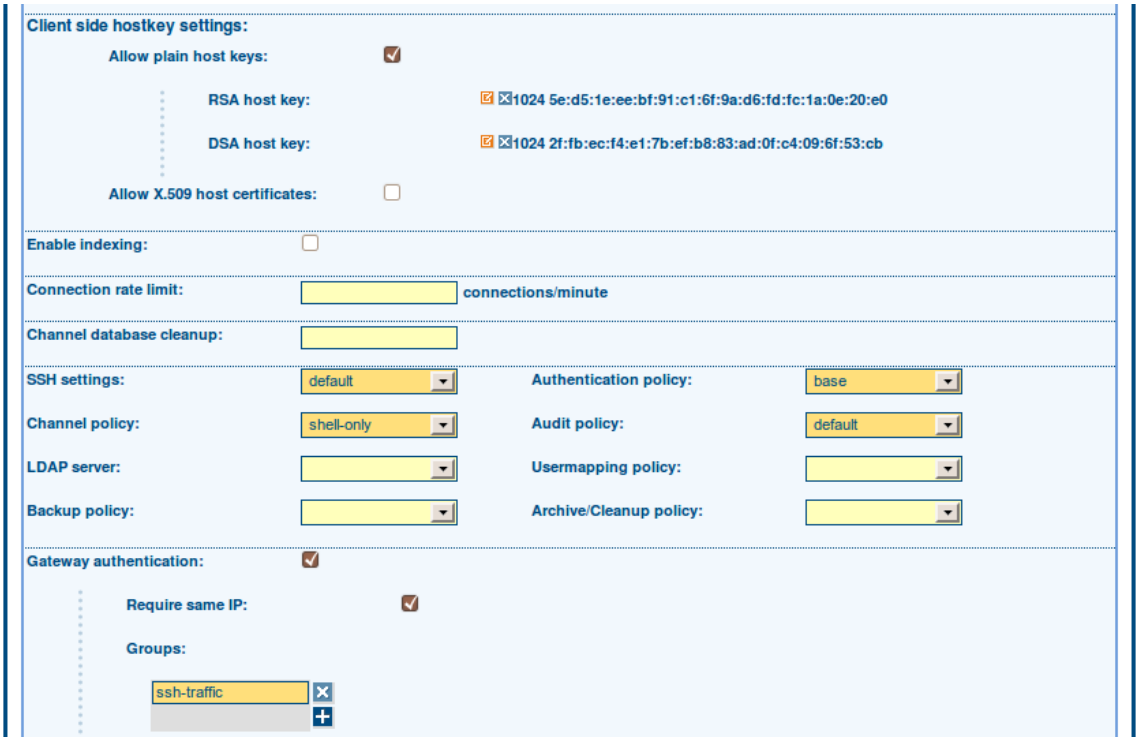


Figure 8.2. Configuring gateway authentication

Select the **Gateway authentication** option.

Step 3. To accept the gateway authentication only from the host that initiated the connection, select **Require same IP**.

Note
This option has no effect if the clients are behind a device that performs network address translation (NAT). In such cases, use inband gateway authentication instead.

Step 4. By default, any user can perform gateway authentication for the connections. To allow only members of a specific group authenticate the connections of this connection policy, select **Groups >** ⊞, and enter the

name of the group whose members can authenticate the connections. This group must exist on the **AAA > Group Management** page. For details on creating and managing usergroups, see *Section 4.4.6, Managing user rights and usergroups (p. 84)*. Repeat this step to add further groups if needed.

Step 5. For SSH and RDP connections, you may want to set a usermapping policy in the **Usermapping policy** field. For details on usermapping policies, see *Procedure 8.1, Configuring usermapping policies (p. 208)*.

Step 6. Click **Commit**. After that, users accessing these connections must perform gateway authentication as described in *Procedure 8.2.2, Performing outband gateway authentication on SCB (p. 212)*.

Step 7. *Optional Step*: To restrict the availability of selected channels of the connection based on the username used for gateway authentication, edit the channel policy used in the connection.

Step a. Navigate to the channel policy used in the connection (for example, **SSH Control > Channel Policies**).

Step b. Select **Gateway Group >** ⊞ and enter the name of the user group allowed to use this type of the channel. The user group must correspond to the username used for the gateway authentication. Repeat this step until all permitted groups are listed.
You may list local user lists as defined in *Procedure 5.1.6, Creating and editing user lists (p. 130)*, or LDAP groups (for details on accessing LDAP servers from SCB, see *Procedure 5.1.7, Authenticating users to an LDAP server (p. 131)*). Note the following behavior of SCB:

- If you list multiple groups, members of any of the groups can access the channel.

Note
When listing both a whitelist and blacklist in the **Gateway Group** section and a username appears on both lists, the user will be able to access the channel.

- If a local user list and an LDAP group has the same name and the LDAP server is configured in the connection that uses this channel policy, both the members of the LDAP group and the members of the local user list can access the channel.

Step c. Click **Commit**.

8.2.2. Procedure – Performing outband gateway authentication on SCB

Steps:

Step 1. Initiate a connection from a client. If gateway authentication is required for the connection, SCB will pause the connection.

Note
For SSH connections, when initiating the connection, you can use the following as your username: `gu=gatewayusername@remoteusername`, where `gatewayusername` is the username you will use to login to the SCB web interface (also called gateway user), and `remoteusername` is the username you will use on the remote server.

Step 2. Open a browser, preferably on the same host you initiated the connection from, and navigate to the login page of SCB.

Warning
If the username used within the protocol is different from the username used to access the SCB web interface to perform gateway authentication, usermapping must be configured for the connection. For details on usermapping, see *Procedure 8.1, Configuring usermapping policies (p. 208)*.

Step 3.

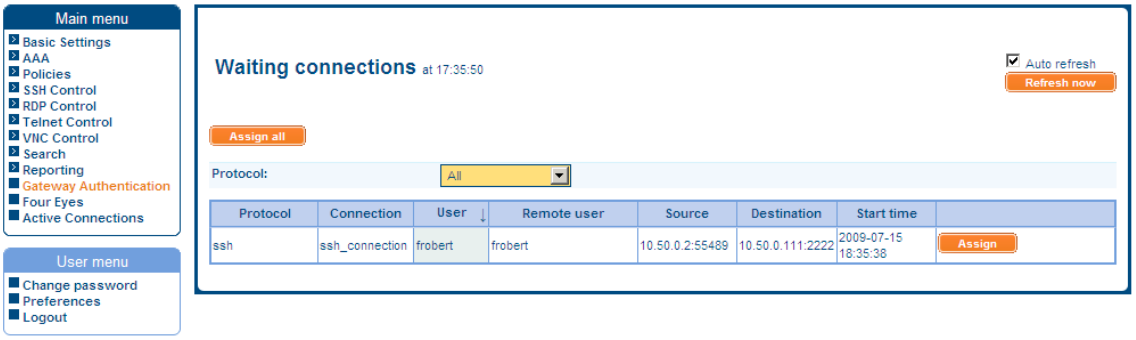


Figure 8.3. Performing gateway authentication

Login to SCB, and select **Gateway Authentication** from the main menu. The list of connections waiting for gateway authentication will be displayed.

Note

- If users accessing the SCB web interface are authenticated to and LDAP server, the users must successfully authenticate to the LDAP server set on the **AAA > Settings** page.
- No other SCB privilege is required to access this page.

Step 4. Select the connection that you started, and click **Assign**.

Step 5. Logout from the SCB web interface.

Step 6. Continue to authenticate on the server.

8.2.3. Procedure – Performing inband gateway authentication in SSH connections

Steps:

Step 1. Initiate a connection from a client. If gateway authentication is required for the connection, SCB will pause the connection.

Step 2. SCB requests the username used for gateway authentication. Enter your gateway username into the **Gateway username:** prompt. If password authentication is used, provide the password for the gateway user as well.

Step 3. The login prompt for the remote server is displayed. Enter your username used on the remote server into the **Username:** prompt. If password authentication is used, provide the password for the username as well.

Warning
If the username used within the protocol is different from the username used to access the SCB web interface to perform gateway authentication, usermapping must be configured for the connection. For details on usermapping, see *Procedure 8.1, Configuring usermapping policies (p. 208)*.

Note
When initiating the connection, you can use the following as your username: `gu=gatewayusername@remoteusername`, where `gatewayusername` is the username you will use to authenticate on SCB and `remoteusername` is the username you will use on the remote server. That way you do not have to provide the usernames in the prompt, only the passwords if password authentication is used.

If SCB is configured to require client-side authentication, the `gatewayusername` user must authenticate on the client side.

8.2.4. Procedure – Performing inband gateway authentication in RDP connections

Steps:

Step 1. Initiate a connection from a client. If gateway authentication is required for the connection, SCB will pause the connection.

Step 2. The graphical login window is displayed.

- If the **Advanced > Terminal Services Gateway > Logon Settings > Use my Terminal Services Gateway credentials for the remote computer** option of your Remote Desktop application is enabled, login to the remote server using your usual credentials. SCB will use these credentials for the gateway authentication on the Domain Controller as well.
- If the **Advanced > Terminal Services Gateway > Logon Settings > Use my Terminal Services Gateway credentials for the remote computer** option of your Remote Desktop application is disabled, first you have to authenticate on the SCB gateway. Enter your username and password for the Domain Controller.
  If the first authentication is successful, a second login window is displayed. Enter your username and password for the remote server you are trying to access.

Step 3. If the authentication is successful, the desktop of the remote server is displayed.

8.2.5. Troubleshooting gateway authentication

If a user initiates a connection and then logs in to the SCB web interface, it might happen that his connection is not shown on the **Gateway Authentication** page. SCB checks the following points to determine if a pending connection is listed for a user:

Warning
The `admin` user is special SCB user and not a member of any user groups; nor can it belong to any group. Since usermapping policies are based on user groups, performing gateway authentication with the `admin` user is likely to result in usermapping errors.

- The username used to access the SCB web interface is a member of a group listed in the **Gateway authentication > Groups** field of the connection policy.
- If SCB knows from the protocol the username that will be used to access the SCB web interface to perform the gateway authentication, the connection is displayed only to this user. Currently this happens only for SSH connections if the client has specified the username for the web interface within the protocol using the *gu=webusername@server-side-username@server* format.
- If the **Gateway authentication > Require same IP** option is enabled, the pending connection is displayed only if the user accesses the SCB web interface from the same IP address as the client in the pending connection.

Note
The *admin* user sees every pending connection.

8.3. Configuring 4-eyes authorization

When 4-eyes authorization is required for a connection, a user (called authorizer) must authorize the connection on SCB as well. This authorization is in addition to any authentication or group membership requirements needed for the user to access the remote server. For details about the concepts of 4-eyes authorization, see *Procedure 2.8, 4-eyes authorization (p. 17)*.

8.3.1. Procedure – Configuring 4-eyes authorization

Purpose:

To require 4-eyes authorization for a connection, complete the following steps.

Steps:

Step 1. Navigate to the **Connections** page of the traffic (for example to **SSH Control > Connections**), and select the connection policy to modify.

Step 2.

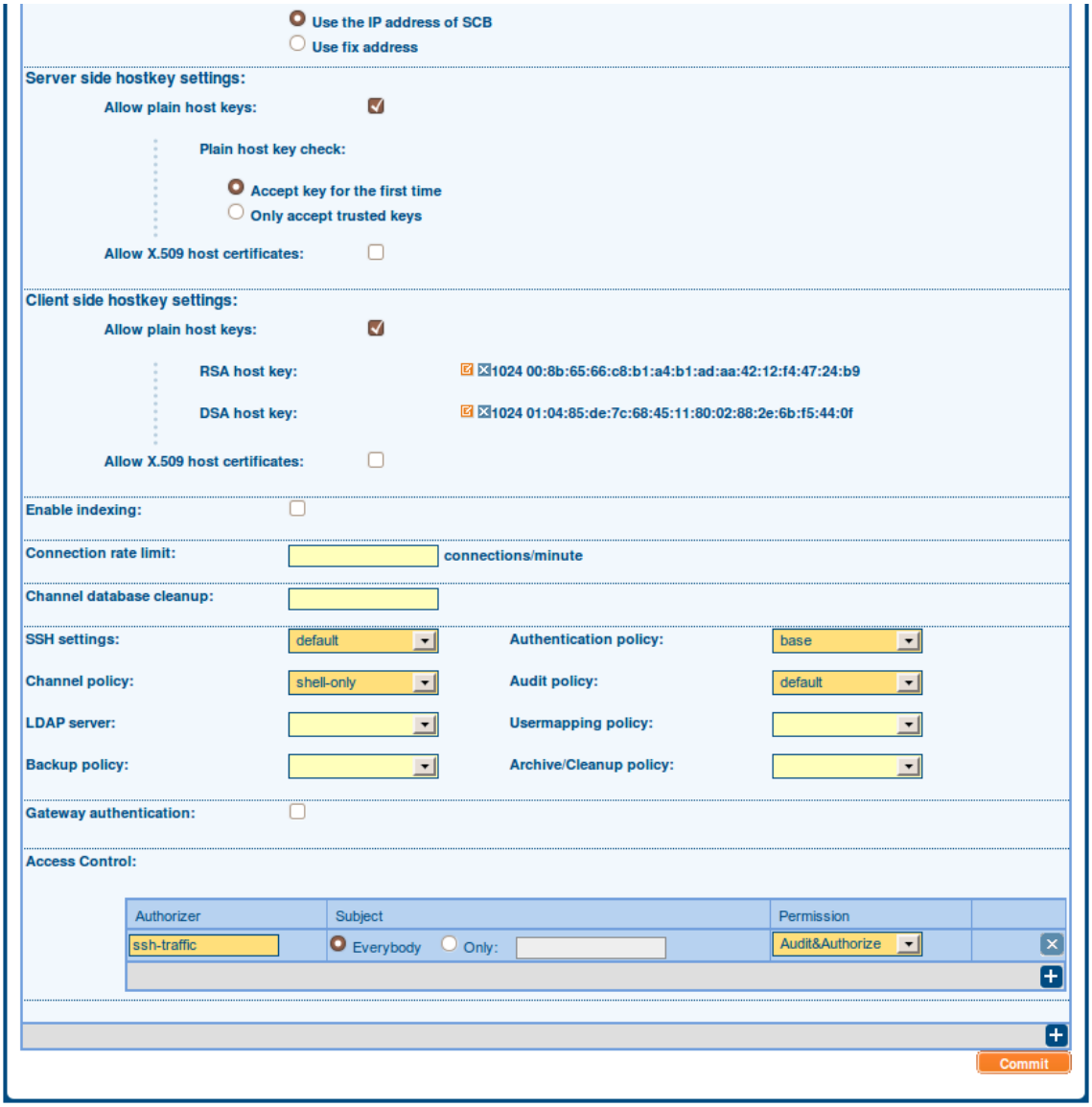


Figure 8.4. Configuring 4-eyes authorization

Navigate to **Access Control** and click ⊞.

Step 3. Enter the name of the usergroup whose members are permitted to authorize the sessions of the connection policy into the **Authorizer** field. This group must exist on the **AAA > Group Management** page. For details on creating and managing usergroups, see *Section 4.4.6, Managing user rights and usergroups (p. 84)*.

Step 4. By default, the authorizer can authorize any session of the connection policy.
If the authorizer is permitted to authorize only the sessions of a certain usergroup, select **Subject > Only**, and enter the name of the userlist whose sessions the authorizer can authorize. If you use 4-eyes authorization without gateway authentication, you can specify an LDAP group instead of a userlist.

Warning
If gateway authentication and 4-eyes authorization is used together in a connection policy, the usergroup of the gateway username must be specified, not the group of the remote username. The specified group must be a local or LDAP group.

Step 5. Set the permissions of the usergroup set in the **Authorizer** field.

- If the **Authorizer** group can authorize (that is, enable) and audit (i,.e monitor in real-time and download the audit trails) the sessions, select **Permission > Audit&Authorize**.
- If the **Authorizer** group can only authorize (that is, enable) the sessions, select **Permission > Authorize**.
- If the **Authorizer** group can only audit (that is, monitor in real-time and download the audit trails) the sessions, select **Permission > Audit**.

Note
If the **Subject > Only** field is set, the auditor can only monitor the sessions of the specified usergroup.

The `admin` user of SCB can audit and authorize every connection.

Step 6. Repeat steps 2-5 to add other authorizers or usergroups if needed.

Step 7. Click **Commit**.

Step 8. Navigate to the **Channel Policies** page of the traffic (for example to **SSH Control > Channel Policies**), and select the channel policy used in the connection.

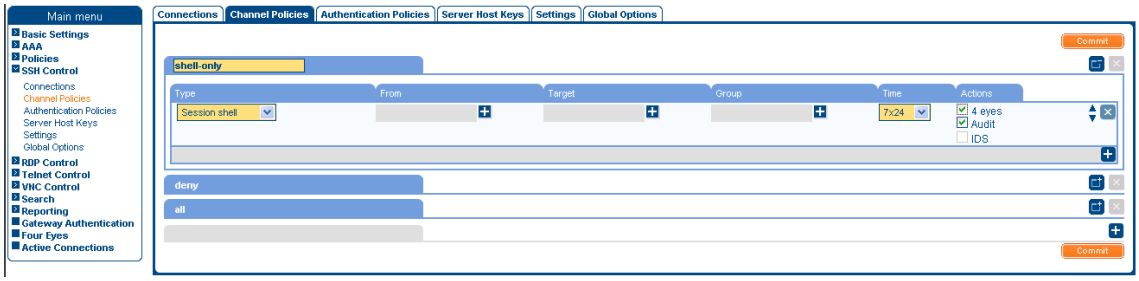


Figure 8.5. Configuring 4-eyes authorization in the channel policy

Step 9. Enable the **4-eyes** option for the channels which should be accessed only using 4-eyes authorization.

Note
If a connection uses secondary channels that require 4-eyes authorization — for example, a Remote Desktop connection allows a Drawing channel but requires 4-eyes authorization for a Disk redirection channel — the connection is locked until the the authorizer accepts the channel on the **Four Eyes** page of SCB. During this time, the client application can become nonresponsive, for example, display the graphical desktop but not react to mouse clicks.

Step 10. Click **Commit**. After that, users accessing connections using the modified channel policy must be authorized as described in *Procedure 8.3.2, Performing 4-eyes authorization on SCB (p. 218)*.

8.3.2. Procedure – Performing 4-eyes authorization on SCB

Steps:

Step 1. When a user initiates a connection from a client and 4-eyes authorization is required for the connection, SCB will pause the connection.

Note
4-eyes authorization can be set separately for every channel. However, if a new channel that requires 4-eyes authorization is opened in an existing connection, the channels already opened are also paused until the 4-eyes authorization is successfully completed.

Step 2.

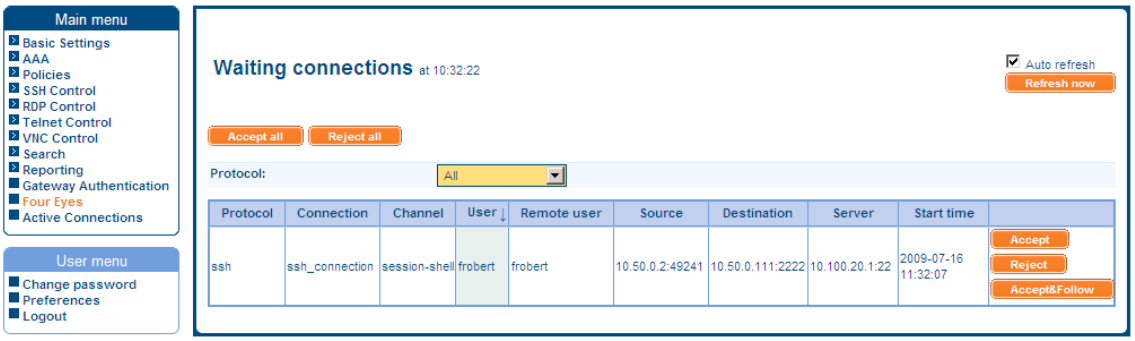


Figure 8.6. Performing 4-eyes authorization

Login to SCB, and select **Four Eyes** from the main menu. The list of connections waiting for authorization will be displayed.

Note
Only those connections will be listed, where your usergroup has the `Authorize` or the `Audit&Authorize` permissions. No other SCB privilege is required to access this page.

Step 3. Select the connection and click **Accept** to enable the connection, **Reject** to deny the connection, or **Accept&Follow** to enable it and monitor in real-time.

Note
Following a session requires the following:

- The **Audit** option must enabled for the specific channel in the Channel policy of the connection.
- The Audit Player application must be installed on the computer of the auditor.
- If the Audit policy of the connection uses encryption, the appropriate decryption keys must be available on the computer of the auditor.

Starting with Audit Player version 2012.1, AP replays the live streams in `live` mode, increasing the replay speed, skipping idle periods, and reverting to real-time replay when reaching the end of the current stream.

Step 4. Enter a note why the connection was accepted/rejected into the appearing dialog box. This description will be stored in the connection database together with other metadata about the connection.

Step 5.

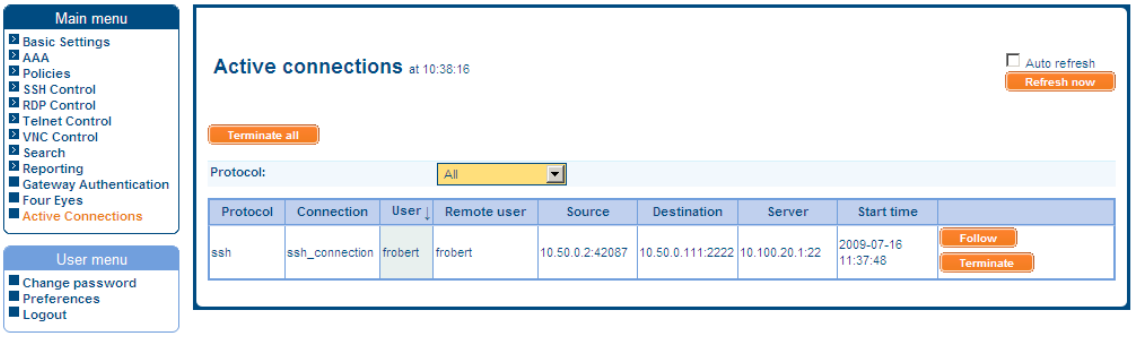


Figure 8.7. Displaying active connections

If you have to terminate an ongoing connection for some reason, select **Active Connections** from the main menu. The list of ongoing connections will be displayed.

Step 6. Select the connection to stop, and click **Terminate**.

Note
When following a connection in the Audit Player application, the auditor can also terminate the connection from the Audit Player by clicking **Terminate**.

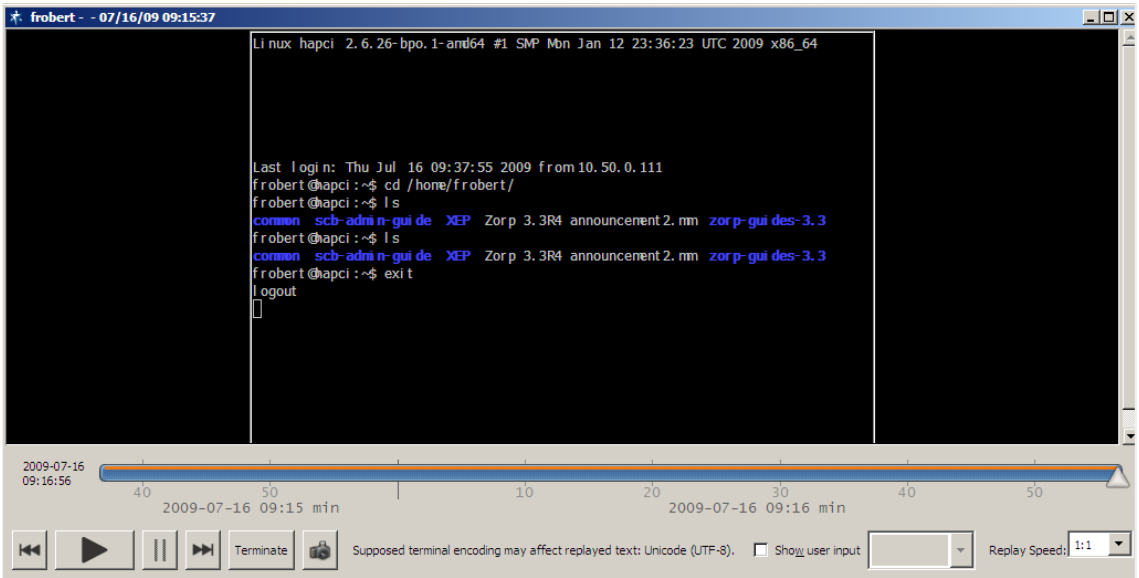

frobert - - 07/16/09 09:15:37
Linux hapci 2.6.26-bpo.1-amd64 #1 SMP Mon Jan 12 23:36:23 UTC 2009 x86_64
Last login: Thu Jul 16 09:37:55 2009 from 10.50.0.111
frobert@hapci:~$ cd /home/frobert/
frobert@hapci:~$ ls
common scb-admin-gui de XEP Zorp 3.3R4 announcement2.mm zorp-gui des-3.3
frobert@hapci:~$ ls
common scb-admin-gui de XEP Zorp 3.3R4 announcement2.mm zorp-gui des-3.3
frobert@hapci:~$ exit
logout
2009-07-16 09:16:56
2009-07-16 09:15 min
2009-07-16 09:16 min
Terminate
Supposed terminal encoding may affect replayed text: Unicode (UTF-8).
Show user input
Replay Speed:
1:1

Figure 8.8. Terminating a connection in AP

Chapter 9. Best practices and configuration examples

This chapter gives step-by-step procedures to configure special aspects of SCB.

9.1. Configuring public-key authentication on SCB

If a protected server requires public-key authentication from the users, complete one of the following procedures.

- In *Procedure 9.1.1, Configuring public-key authentication using local keys (p. 220)*, SCB stores the public keys of the users and the private-public keypair used in the server-side connection locally on SCB.
- In *Procedure 9.1.2, Configuring public-key authentication using an LDAP server and a fixed key (p. 221)*, SCB receives the public keys of the users from an LDAP server and uses a locally-stored private-public keypair in the server-side connection.
- In *Procedure 9.1.3, Configuring public-key authentication using an LDAP server and generated keys (p. 222)*, SCB receives the public keys of the users from an LDAP server. SCB generates a keypair that is used in the server-side connection on-the-fly, then uploads the public key of this pair to the LDAP database. That way the server can authenticate SCB to the (newly generated) public key of the user.

9.1.1. Procedure – Configuring public-key authentication using local keys

Purpose:

To store the public keys of the users and the private-public keypair used in the server-side connection locally on SCB, complete the following steps:

Steps:

Step 1. Navigate to **SSH Control > Authentication Policies** and create a new Authentication Policy.

Step 2. Select **Client authentication backend > Local > Publickey**, deselect all other options.

Step 3. Select **Server authentication methods > Publickey > Map**, deselect all other options.

Step 4. Enter the client's username used on the server into the **Username** field.

Step 5. Click the icon in the **Client Side** field. A popup window is displayed.

Step 6. Click **Browse**, select the public key of the user, then click **Upload**. Alternatively, you can paste the key into the **Copy-paste** field and click **Upload**.

Step 7. Click the icon in the **Server Side** field. A popup window is displayed.

Step 8. Click **Browse** and select the private key of the user, or paste the key into the **Copy-paste** field. Enter the password for the private key into the **Password** field and click **Upload**.
If the private key of the user is not available, click **Generate** to create a new private key. You can set the size of the key in the **Generate key** field.In this case, do not forget to export the public key from SCB and import it to the server. To export the key from SCB, just click on the key and save it to your local computer.

Step 9. Repeat Steps 3-8 to add other users.

Step 10. Click **Commit**.

Step 11. Navigate to **SSH Control > Connections** and create a new Connection.

Step 12. Enter the IP addresses of the clients and the servers into the **From** and **To** fields.

Step 13. Select the authentication policy created in Step 1 in the **Authentication Policy** field.

Step 14. Configure the other options of the connection as necessary.

Step 15. Click **Commit**.

Step 16. To test the above settings, initiate a connection from the client machine to the server.

9.1.2. Procedure – Configuring public-key authentication using an LDAP server and a fixed key

Purpose:

To fetch the public keys of the users from an LDAP server and use a locally-stored private-public keypair in the server-side connection, complete the following steps:

Steps:

Step 1. Navigate to **SSH Control > Authentication Policies** and create a new Authentication Policy.

Step 2. Select **Client authentication backend > Local > Publickey**, deselect all other options.

Step 3. Select **Server authentication methods > Publickey > Fix**, deselect all other options.

Step 4. Select **Private key >** . A popup window is displayed.

Step 5. Click **Browse** and select the private key of the user, or paste the key into the **Copy-paste** field. Enter the password for the private key into the **Password** field and click **Upload**.
If the private key of the user is not available, click **Generate** to create a new private key. You can set the size of the key in the **Generate key** field.In this case, do not forget to export the public key from SCB and import it to the server. To export the key from SCB, just click on the key and save it to your local computer.

Step 6. Click on the fingerprint of the key in the **Server side (private-public keypair)** field and save the public key. Do not forget to import this public key to the server: all connections that use this new authentication policy will use this keypair on the server side.

Step 7. Click **Commit**.

Step 8. Navigate to **Policies > LDAP Servers** and click to create a new LDAP policy.

Step 9. Enter the parameters of the LDAP server. For details, see *Procedure 5.1.7, Authenticating users to an LDAP server (p. 131)*.

Step 10. If different from `sshPublicKey`, enter the name of the LDAP attribute that stores the public keys of the users into the **Publickey attribute name** field.

Warning
The public keys stored in the LDAP database must be in OpenSSH format.

Step 11. Navigate to **SSH Control > Connections** and create a new Connection.

Step 12. Enter the IP addresses of the clients and the servers into the **From** and **To** fields.

Step 13. Select the authentication policy created in Step 1 from the **Authentication Policy** combobox.

Step 14. Select the LDAP policy created in Step 7 from the **LDAP Server** combobox.

Step 15. If the server accepts a user only from a specific IP address, select the **Use original IP address of the client** radiobutton from the **SNAT** field.

Step 16. Configure the other options of the connection as necessary.

Step 17. Click **Commit**.

Step 18. To test the above settings, initiate a connection from the client machine to the server.

9.1.3. Procedure – Configuring public-key authentication using an LDAP server and generated keys

Purpose:

To fetch the public keys of the users from an LDAP server and have SCB generate a keypair that is used in the server-side connection on-the-fly, and upload the public key of this pair to the LDAP database, complete the following steps:

Steps:

Step 1. Navigate to **SSH Control > Authentication Policies** and create a new Authentication Policy.

Step 2. Select **Client authentication backend > Local > Publickey**, deselect all other options.

Step 3. Select **Server authentication methods > Publickey > Publish to LDAP**, deselect all other options.

Step 4. Click **Commit**.

Step 5. Navigate to **Policies > LDAP Servers** and click ⊞ to create a new LDAP policy.

Step 6. Enter the parameters of the LDAP server. For details, see *Procedure 5.1.7, Authenticating users to an LDAP server (p. 131)*.

Step 7. If different from `sshPublicKey`, enter the name of the LDAP attribute that stores the public keys of the users into the **Publickey attribute name** field.

Warning
The public keys stored in the LDAP database must be in OpenSSH format.

Step 8. Enter the name of the LDAP attribute where SCB shall upload the generated keys into the **Generated publickey attribute name** field.

Step 9. Click **Commit**.

Step 10. Navigate to **SSH Control > Connections** and create a new Connection.

Step 11. Enter the IP addresses of the clients and the servers into the **From** and **To** fields.

Step 12. Select the authentication policy created in Step 1 from the **Authentication Policy** combobox.

Step 13. Select the LDAP policy created in Step 7 from the **LDAP Server** combobox.

Step 14. If the server accepts a user only from a specific IP address, select the **Use original IP address of the client** radiobutton from the **SNAT** field.

Step 15. Configure the other options of the connection as necessary.

Step 16. Click **Commit**.

Step 17. To test the above settings, initiate a connection from the client machine to the server.

9.2. Organizing connections in Bastion mode

When using SCB in Bastion mode, the administrators must address the external interface of SCB to access the protected servers. If an administrator has access to more than one protected server, SCB must be able to determine which server the administrator wants to access. For each protected server, the administrators must address either different ports of the external interface, or different alias IP addresses of the external interface.

9.2.1. Procedure – Organizing connections based on port numbers

Purpose:

To organize connections based on port numbers, complete the following steps:

Organizing connections based on port numbers is advantageous if the external interface of SCB has a public IP address and the protected servers must be administered from the Internet.

Steps:

Step 1. Navigate to the **Connections** tab of the **Secure Shell Control** menu.

Step 2. Add a new connection. Enter the IP address of the administrators into the **From** fields, and the IP address and port number of the server into the **Target** field.

Step 3. Enter the IP address of the external interface of SCB into the **To** field, and enter a port number into the **Port** field.

Step 4. Repeat Steps 2-3 for every protected server, but every time use a different port number in Step 3.

Warning
Do not use ports *22* and *443* if the management interface of SCB is not configured. Using these ports may make the web interface and the SSH server of SCB unaccessible.

Step 5. Click **Commit**. The administrators can access the protected servers by connecting to the IP address of SCB's external interface, and use the port number to select which server they want to access.

9.2.2. Procedure – Organizing connections based on alias IP addresses

Purpose:

Organizing connections based on alias IP addresses is advantageous if the external interface of SCB is connected to a private network and many private IP addresses are available.

Steps:

Step 1. Navigate to the **Network** tab of the **Basic Settings** menu.

Step 2. Click ⊞ in the **External interface** section to add a a new alias IP address to the external interface. Repeat this step for every protected server. Use a different IP address each time.

Step 3. Navigate to the **Connections** tab of the **Secure Shell Control** menu.

Step 4. Add a new connection. Enter the IP address of the administrators into the **From** fields, and the IP address and port number of the server into the **Target** field.

Step 5. Enter an alias IP address of the external interface of SCB into the **To** field.

Warning
Do not use the primary IP address of the external interface if the management interface of SCB is not configured. Using this IP address may make the SSH server of SCB unaccessible.

Step 6. Repeat Steps 4-5 for every protected server, but every time use a different alias IP address in Step 5.

Step 7. Click **Commit**. The administrators can access the protected servers by connecting to an alias IP address of SCB's external interface. The alias IP address determines which server they will access.

9.2.3. Accessing the SCB host in Bastion mode using SSH

If the management interface is not configured, the SCB host can be accessed using SSH only on port 22 of the external interface. But if the primary IP address of the external interface and port 22 is used in a connection policy, the SSH connection targeting SCB is redirected to a protected server, and SCB cannot be accessed using SSH.

Do not use port 22 of the external interface in connection policies: use other port numbers or alias IP addresses instead. For details, see *Procedure 9.2.1, Organizing connections based on port numbers (p. 223)* and *Procedure 9.2.2, Organizing connections based on alias IP addresses (p. 223)*. Use these methods if possible.

For details on enabling the SSH server of SCB, see *Procedure 4.5.4.2, Enabling SSH access to the SCB host (p. 111)*.

9.3. Procedure – Using nontransparent Bastion mode

Purpose:

Nontransparent mode allows you to create a single connection policy and allow users to access any server by including the name of the target server in their username (for example `ssh username@targetserver@scb_address`). To configure nontransparent mode, complete the following steps.

Note
Nontransparent mode is most commonly used in Bastion mode, but it works with other modes of operation as well.

Steps:

Step 1. Select the type of connection from the main menu.
To configure a Secure Shell connection, select **Secure Shell Control**.

To configure a Remote Desktop connection, select **RDP Control**.

Note
Nontransparent mode in the Telnet protocol is currently not supported.

Step 2. Click and enter a name that will identify the connection (for example `admin_mainserver`).

Step 3. Enter the IP address of the client that will be permitted to access the server into the **From** field. Click to list additional clients.

Step 4. Enter the IP address of SCB's external interface into the **To** field.

Warning
When using SSH connections in nontransparent mode, the clients should not address port 22 of SCB's external interface, as it may make SCB unaccessible for remote maintenance. For details, see *Section 9.2.3, Accessing the SCB host in Bastion mode using SSH (p. 224)*.

Step 5. Click to display the details of the connection and select **Inband destination selection**.

Step 6. Enter the IP address or the hostname of the domain name server used to resolve the address of the server extracted from the username into the **DNS Server** field.

Step 7. If the clients do not include the domain name when addressing the server (for example they use `username@server` instead of `username@server1.example.com`), SCB can automatically add domain information (for example `example.com`). Enter the domain name to add into the **DNS Suffix** field.

Note
If the **DNS Suffix** field is filled, SCB will always add this suffix to the hostname. If you specify a DNS suffix, make sure that the clients do not include the domain name in their requests.

Tip
Windows RDP clients often send only the first 9 characters of the username to the server, making nontransparent operation difficult. The DNS Suffix parameter can be used to complete the missing part. Also, it is not necessary to include the username when starting an RDP connection (for example use only `@server`); the user can type the username later in the graphical login screen.

Step 8. Click and enter the addresses of the servers that the users are permitted to access into the **Targets** field. Use the IP address/netmask (for example `192.168.2.16/32`) format, or enter the hostname of the server, as addressed by the client. The hostnames may contain the `*` and `?` wildcard characters. Do not include the DNS Suffix in the hostname if you have set it in the previous step.

If the clients address the server using its IP address, make sure to include the IP address of the server in the Targets list. This is required because SCB resolves the hostnames to IP addresses, but does not reverse-resolve IP addresses to hostnames. So if only the hostname of the server is listed and the client addresses the server using its IP address, SCB will refuse the connection.

Step 9. Leave the **Port** field empty if the clients may access any port on the server, or enter the specified port they can access.

Step 10. Configure other parameters of the connection as needed.

Step 11. Click **Commit**.

9.4. Procedure – Restoring SCB configuration and data

Purpose:

The following procedure describes how to restore the configuration and data of SCB from a complete backup, for example, after a hardware replacement.

Steps:

Step 1. Connect to your backup server and locate the directory where SCB saves the backups. The configuration backups are stored in the `config` subdirectory in timestamped files. Find the latest configuration file (the configuration files are called *`SCB-timestamp.config`*).

Step 2. Connect to SCB.
If you have not yet completed the Welcome Wizard, click **Browse**, select the configuration file, and click **Import**.

If you have already completed the Welcome Wizard, navigate to **Basic Settings > System > Import configuration > Browse**, select the configuration file, and click **Import**.

Step 3. Navigate to **Policies > Backup & Archive/Cleanup**. Verify that the settings of the target servers and the backup protocols are correct.

Step 4. Navigate to **Basic Settings > Management > System backup**, click **Restore now** and wait for the process to finish. Depending on the amount of data stored in the backup and the speed of the connection to the backup server, this may take a long time.

Step 5. Navigate to **SSH Control > Connections**, and click **Restore ALL**. Repeat this step for other traffic types. Depending on the amount of data stored in the backup and the speed of the connection to the backup server, this may take a long time.

Chapter 10. SCB scenarios

This chapter discusses common scenarios for SCB.

10.1. Procedure – SSH usermapping and keymapping in AD with public key

Purpose:

A customer wants to be able to disable password authentication in SSH for admin users on the UNIX servers. However, the customer uses Active Directory, and would not like to enter the username/password at gateway authentication for every login over and over again. Therefore, the customer needs a quasi SSO-like system, with only one group logging in as `root` and another group as `XY` user.

Steps:

Step 1. Create an LDAP authentication policy. For details on creating a new authentication policy, see *Section 5.2.3, Authentication Policies (p. 154)*. In this scenario, only a few important details will be highlighted.

1. In the **Client authentication backend** field, set the authentication method used on the client-side to **LDAP**. This will be the Active Directory where the gateway will get the public key from, for authentication. Enable **Publickey** only from the **Authentication methods** and disable all other methods.
2. In the **Server authentication methods** field, enable **Publickey** and disable all other methods. Under **Publickey**, set the **Server side private and public key** to **Map** so that end users will not know the key.
3. Navigate to the bottom of the policy, and click .
4. Enter the username in the **Username** field (for example: `root`). Generate a **Private key** and upload its public counterpart to the server.

Step 2. Set an LDAP server policy where you setup the active directory. For details on authenticating users to an LDAP server, see *Procedure 5.1.7, Authenticating users to an LDAP server (p. 131)*. In this scenario, only a few important details will be highlighted.

1. If the domain name is `DEMO.balabit`, then enter the following:
2. **Base DN:** `DC=DEMO,DC=balabit`
3. **Bind DN:** `CN=Administrator,CN=Users,DC=DEMO,DC=balabit`
4. **Bind Password:** `<the password of the admin>`
5. **Publickey attribute name:**: `sshPublicKey`

Step 3. By default, the active directory does not have any attribute that could store the SSH public key. To solve this, perform the following steps:

1. Enable Schema updates by means of the registry:
   a. Click **Start**, click **Run**, and then in the **Open** box, type: `regedit`. Press **Enter**.
   b. Locate and click the following registry key: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters**.

c. On the **Edit** menu, click **New**, and then click **DWORD Value**.

d. Enter the value data when the following registry value is displayed:

```
Value Name: Schema Update Allowed
Data Type: REG_DWORD
Base: Binary
Value Data: 1
```

Note
Type *1* to enable this feature, or *0* (zero) to disable it.

e. Quit Registry Editor.

2. To install the Schema snap-in, click **Start**, click **Run**, and then in the **Open** box, type: *regsvr32 schmmgmt.dll*. Press **Enter**.

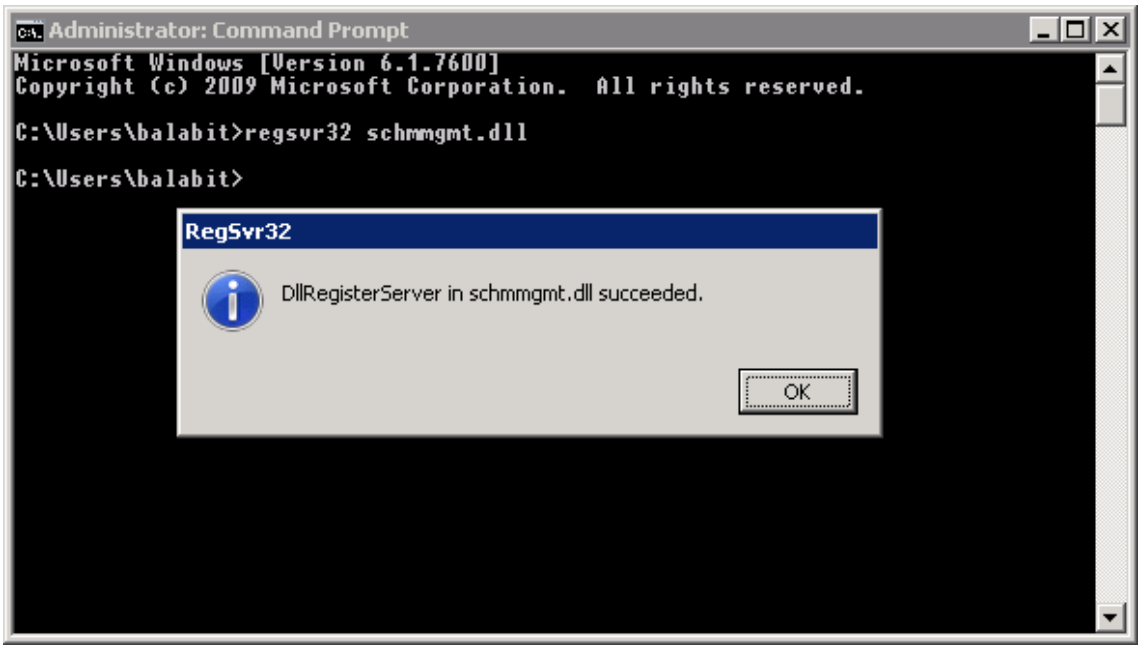


Figure 10.1. Installing the Schema snap-in

3. Click **Start**, click **Run**, and then in the **Open** box, type: *MMC*. Press **Enter**.
4. Navigate to **File > Add or Remove Snap-in**, select **Active Directory Schema** and click **Add**.

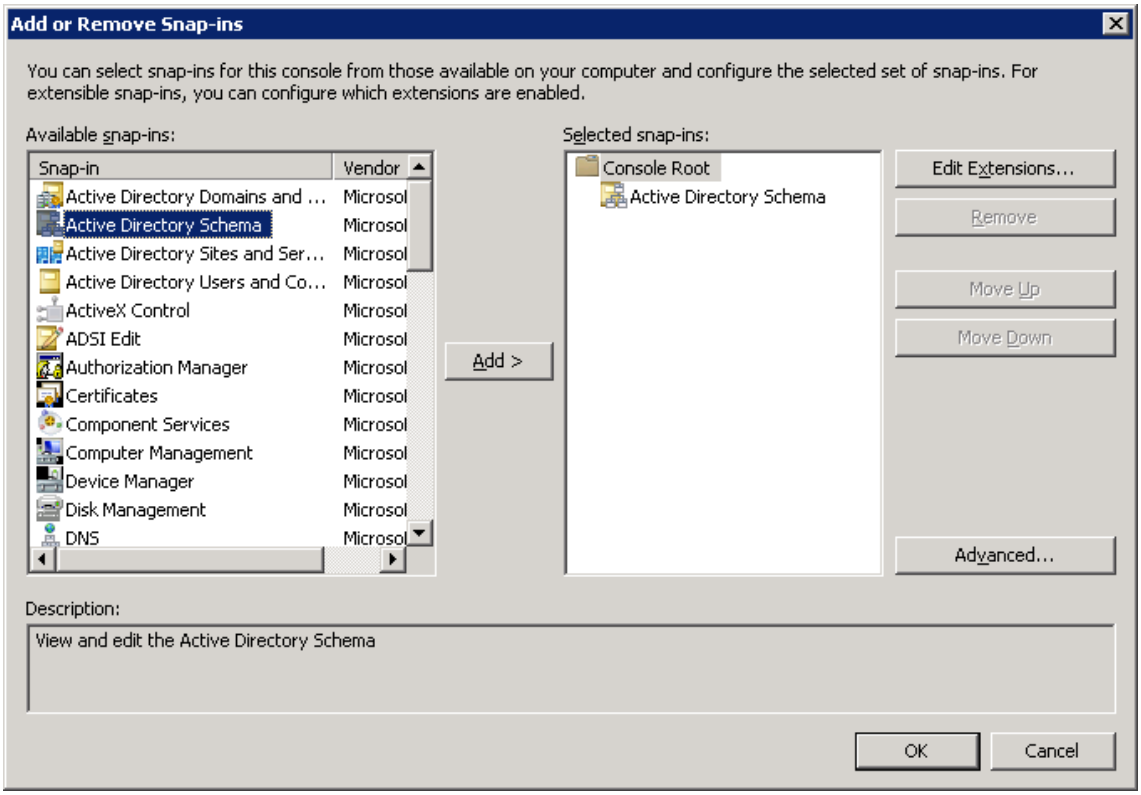


Figure 10.2. Add or Remove Snap-in

5. Expand the Active Directory schema and right click **Attributes**.
6. Click **Create Attribute**. If a warning appears, click **Continue**.

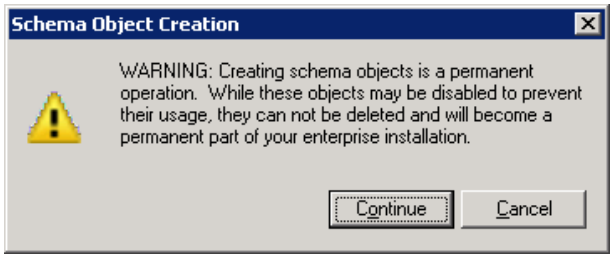

Schema Object Creation
WARNING: Creating schema objects is a permanent operation. While these objects may be disabled to prevent their usage, they can not be deleted and will become a permanent part of your enterprise installation.
Continue
Cancel

Figure 10.3. Creating Attribute - Warning

7. In **Common name** and **LDAP name** enter `sshPublicKey`.
8. In **OID** enter `1.3.6.1.4.1.24552.1.1.1.13`.
9. For **Syntax**, select `IA5-String`.
10. Enable **Multi-Valued**.

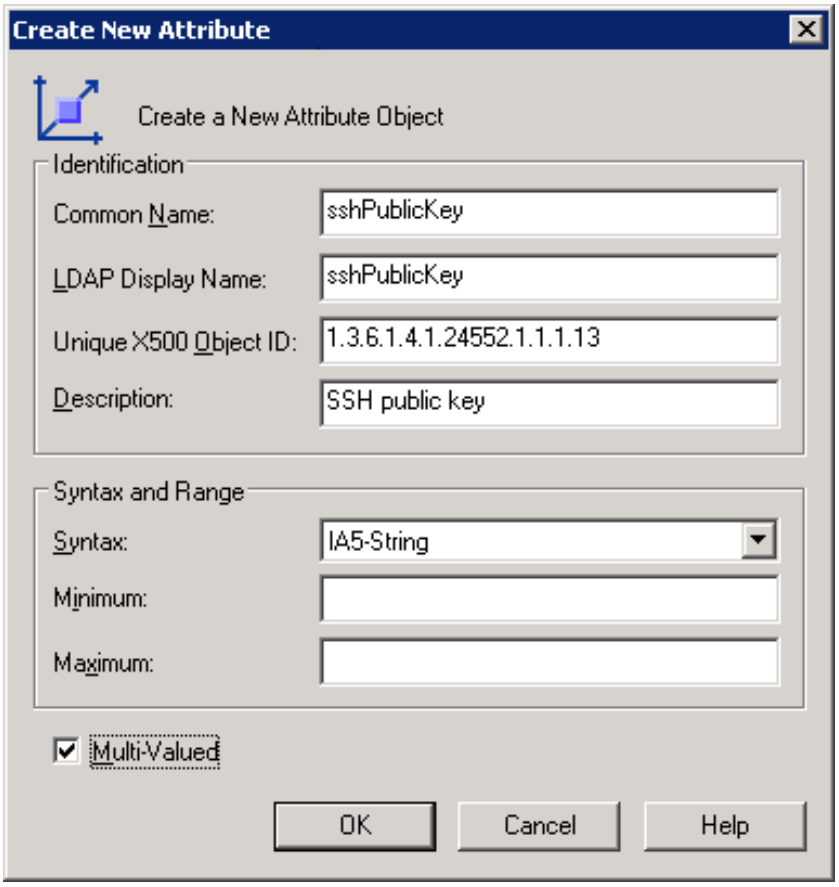


Figure 10.4. Create New Attribute

11. Right click **Classes** and click **Create class**. If a warning appears, click **Continue**.
12. Name the class as `ldapPublicKey`.
13. In **X500 OID** enter `1.3.6.1.4.1.24552.500.1.1.2.0`.
14. In **Parent Class** enter `top`, and in **Class Type** enter `Auxiliary`.

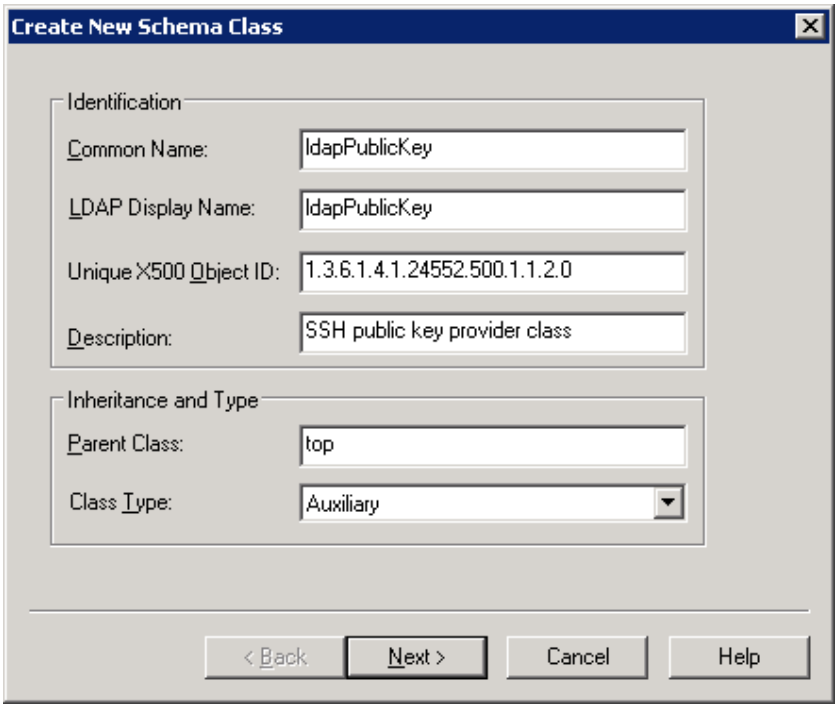


Figure 10.5. Create New Schema Class

15. Click **Next**. Add `sshPublicKey` to the **Optional** field. Click **Finish**.

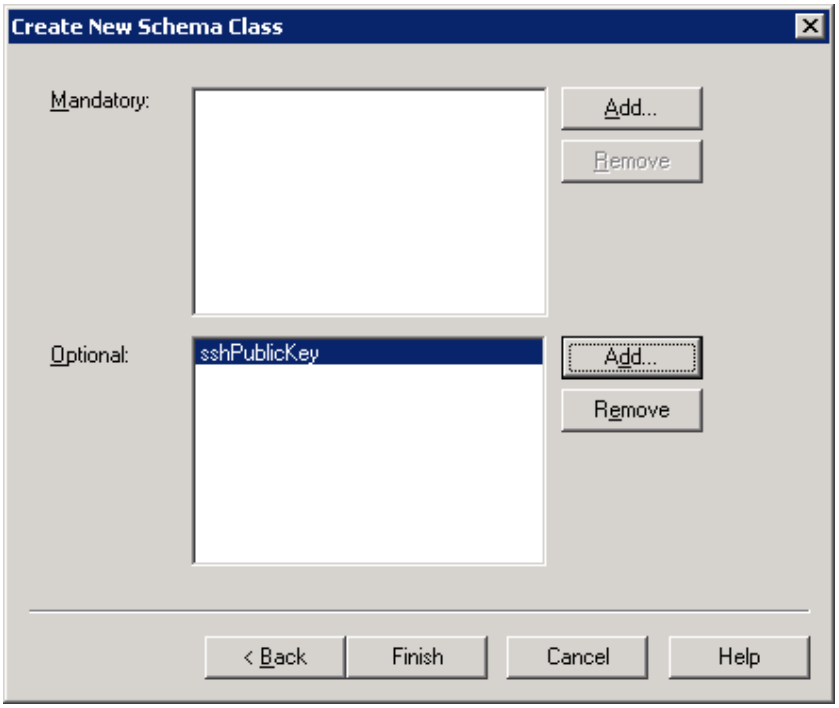


Figure 10.6. Create New Schema Class (Optional)

16. Expand **Classes** and select **User**. Right click **User** and select **Properties**. Navigate to **Relationship > Auxiliary class**, click **Add Class** and add the **ldapPublicKey** class. Click **Apply**.

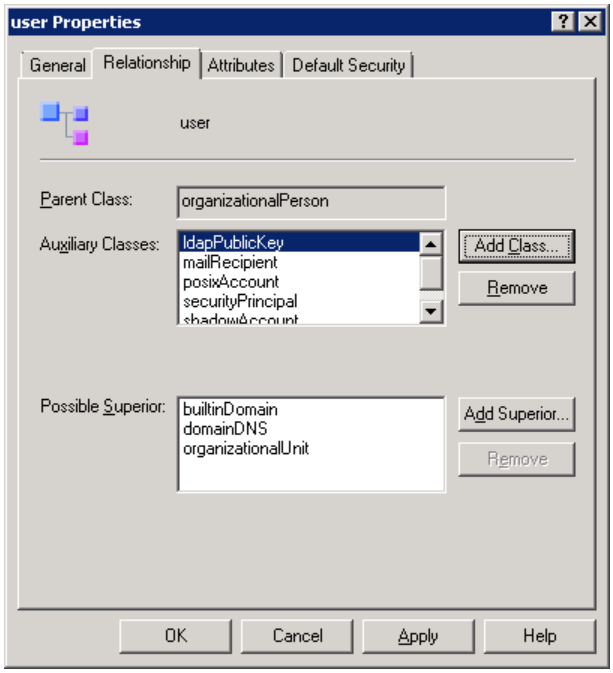


Figure 10.7. User Properties

Step 4. The next step is to map the public keys to users. This is not possible in a user editor, use a low-level LDAP utility instead.

1. Add ADSI Edit as a snap-in to MMC.

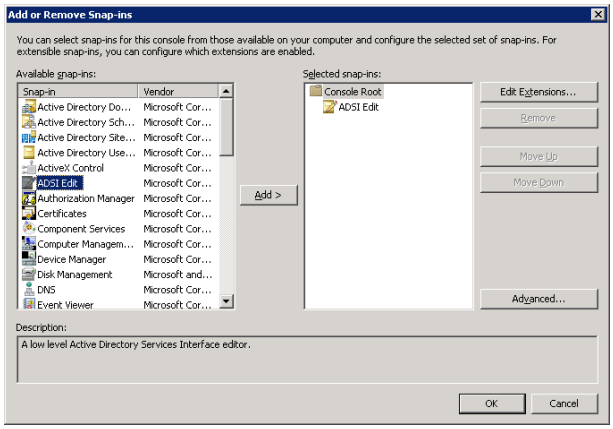


Figure 10.8. Adding ADSI Edit

2. Right-click on the node and press **Enter**.
3. Search for the user in the tree, right-click on it and select **Properties.** All attributes can be edited there, so sshPublicKey also. Add the public keys to openssh one by one.

Step 5. Create a usermapping policy where you will set those groups from the Active Directory who can become root. For details on creating usermapping policies, see *Procedure 8.1, Configuring usermapping policies (p. 208)*. In this scenario, only a few important details will be highlighted. Set the **Username** to `root` and select

the group you intend to give these rights to. If you intend to allow other users in without usermapping, enable **Allow other unmapped usernames**.

Chapter 11. Troubleshooting SCB

This section describes the tools to detect networking problems, and also how to collect core files and view the system logs of SCB.

11.1. Procedure – Network troubleshooting

Purpose:

The **Troubleshooting** menu provides a number of diagnostic commands to resolve networking issues. Logfiles of SCB can also be displayed here — for details, see *Procedure 11.3, Viewing logs on SCB (p. 236)*.

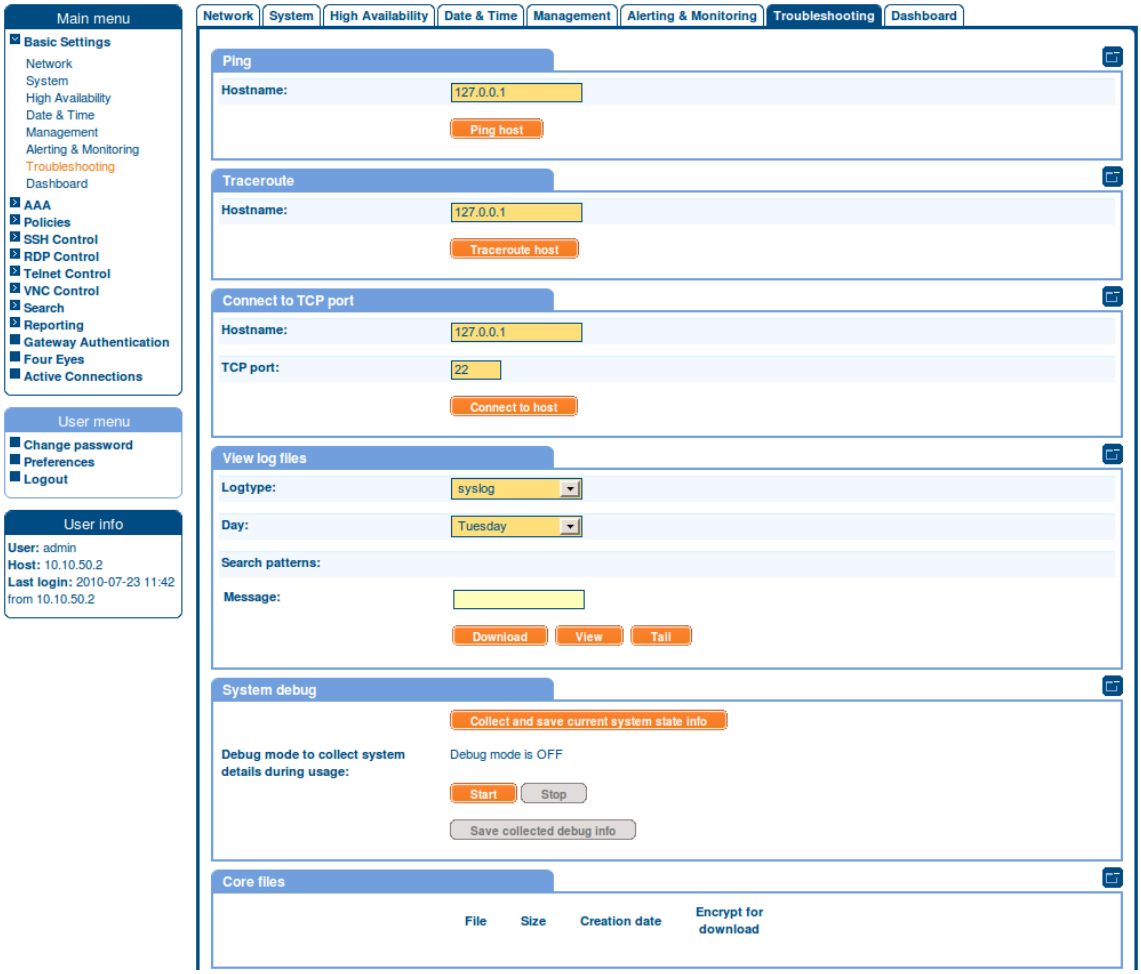


Figure 11.1. Network troubleshooting with SCB

The following commands are available:

- *ping*: Sends a simple message to the specified host to test network connectivity.

- *traceroute*: Sends a simple message from SCB to the specified host and displays all hosts on the path of the message. It is used to trace the path the message travels between the hosts.
- *connect*: Attempts to connect the specified host using the specified port. It is used to test the availability or status of an application on the target host.

To execute one of the above commands, complete the following steps:

Steps:

Step 1. Navigate to **Basic Settings > Troubleshooting**.

Step 2. Enter the IP address or the hostname of the target host into the **Hostname** field of the respective command. For the `Connect` command, enter the target port into the **Port** field.

Step 3. Click the respective action button to execute the command.

Step 4. Check the results in the popup window. Log files are displayed in a separate browser window.

11.2. Gathering data about system problems

SCB automatically generates core files if an important software component (for example Zorp) of the system crashes for some reason. These core files can be of great help to the BalaBit Support Team to identify problems. When a core file is generated, the SCB administrator receives an alerting e-mail, and an SNMP trap is generated if alerting is properly configured (for details, see *Section 4.3.4, Configuring system monitoring on SCB (p. 57)* and *Section 4.3.3, System logging, SNMP and e-mail alerts (p. 52)*). To display a list of alerts if monitoring is not configured, navigate to **Search > Log Alerts**.

To list and download the generated core files, navigate to **Basic Settings > Troubleshooting > Core files**.

By default, core files are deleted after 14 days. To change the deletion timeframe, navigate to **Basic Settings > Management > Core files**.

Warning
In the current version of SCB, core files are automatically deleted if the system is rebooted.

11.3. Procedure – Viewing logs on SCB

Purpose:

The **Troubleshooting** menu provides an interface to view the logs generated by the various components of SCB.

Note
Because of performance reasons, log files larger than 2 Megabytes are not displayed in the web interface. To access these logs, download the file instead.

Steps:

Step 1. Navigate to **Basic Settings > Troubleshooting > View log files**.

Step 2. Use the **Logtype** roll-down menu to select the message type.

- *SCB*: Logs of the SCB web interface.
- *syslog*: All system logs of the SCB host.
- *ssh*: Logs of the SSH connections passing SCB.
- *rdp*: Logs of the RDP connections passing SCB.
- *telnet*: Logs of the Telnet connections passing SCB.
- *vnc*: Logs of the VNC connections passing SCB.

Step 3.

- To download the log file, click **Download**.
- To follow the current log messages real-time, click **Tail**.
- To display the log messages, click **View**.

Step 4. To display log messages of the last seven days, select the desired day from the **Day:** field and click **View**.

Tip
To display only the messages of a selected host or process, enter the name of the host or process into the **Message:** field.

The **Message:** field acts as a generic filter: enter a keyword or a regular expression to display only messages that contain the keyword or match the expression.

11.4. Procedure – Changing log verbosity level of SCB

Purpose:

The logging level of SCB can be set separately for every protocol. To change the verbosity level of SCB, complete the following steps:

Note
The **Basic Settings > Management > Debug logging > Enable debug logs** option is not related to the verbosity of traffic logs: it increases the log level of the non-network-related events, for example adds the commands executed by the SCB web interface to the logs, and so on

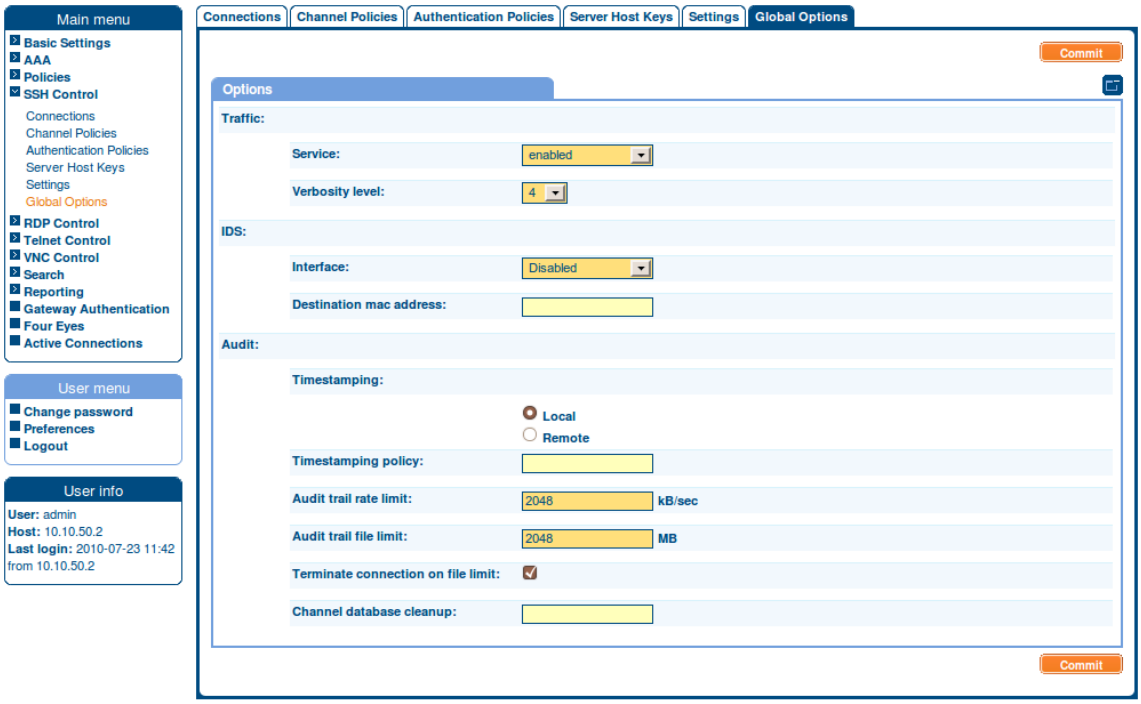


Figure 11.2. Changing the verbosity level

Steps:

Step 1. Navigate to the **Global Options** page of the traffic you want to change the log level of; for example to **SSH Control > Global Options** to change the log level of SSH traffic, **RDP Control > Global Options** for remote desktop traffic, and so on

Step 2. Select the desired log level from the **Verbosity level** field.

Note
The verbosity level ranges from 1 (no logging) to 10 (extremely detailed), with level 4 being the default normal level. To debug complex problems, you might have to increase the verbosity level to 6. Higher level is needed only in extreme cases.

Note that high verbosity levels generate very large amount of log messages.

11.5. Procedure – Collecting logs and system information for error reporting

Purpose:

To track down support requests, the BalaBit Support Team might request you to collect system-state and debugging information. This information is collected automatically, and contains log files, the configuration file of SCB, and various system-statistics.

Note
Sensitive data like key files and passwords are automatically removed from the files.

The **Basic Settings > Management > Debug logging > Enable debug logs** option is not related to the verbosity of log messages: it adds the commands executed by the SCB web interface to the log.

To collect system-state information, navigate to **Basic Settings > Troubleshooting > System debug** and click **Collect and save current system state info**, then save the created zip file. The name of the file uses the `debug_info-<hostname>YYYYMMDDHHMM` format.

To collect information for a specific error, complete the following steps:

Steps:

Step 1. Navigate to **Basic Settings > Troubleshooting > System debug**.

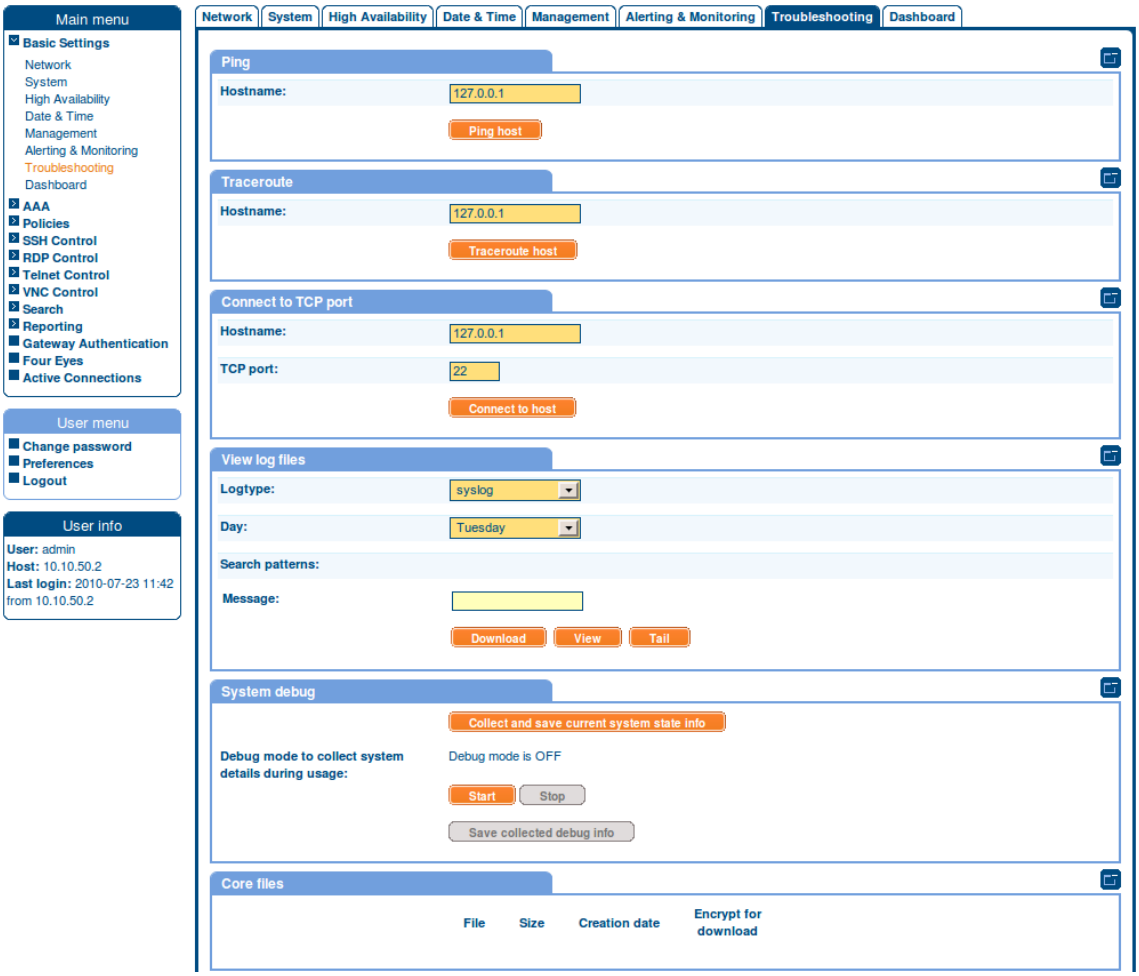


Figure 11.3. Collecting debug information

Step 2. Click **Start**.

Note
Starting debug mode increases the log level of SCB, and might cause performance problems if the system is under a high load.

Step 3. Reproduce the event that causes the error, for example connect to a server.

Step 4. Click **Stop**.

Step 5. Click **Save the collected debug info** and save the created zip file. The name of the file uses the *debug_info-<hostname>YYYYMMDDHHMM* format.

Step 6. Attach the file to your support ticket.

11.6. Status history and statistics

SCB displays various statistics and status history of system data and performance on the dashboard at **Basic Settings > Dashboard**. The dashboard is essentially an extension of the system monitor: the system monitor displays only the current values, while the dashboard creates graphs and statistics of the system parameters.

The dashboard consists of different modules. Every module displays the history of a system parameter for the current day. To display the graph for a longer period (last week, last month, or last year), select the **Week**, **Month**, or **Year** options, respectively. Hovering the mouse over a module enlarges the graph and displays the color code used on the graph.

To display statistics of a module as a table for the selected period, click on the graph.

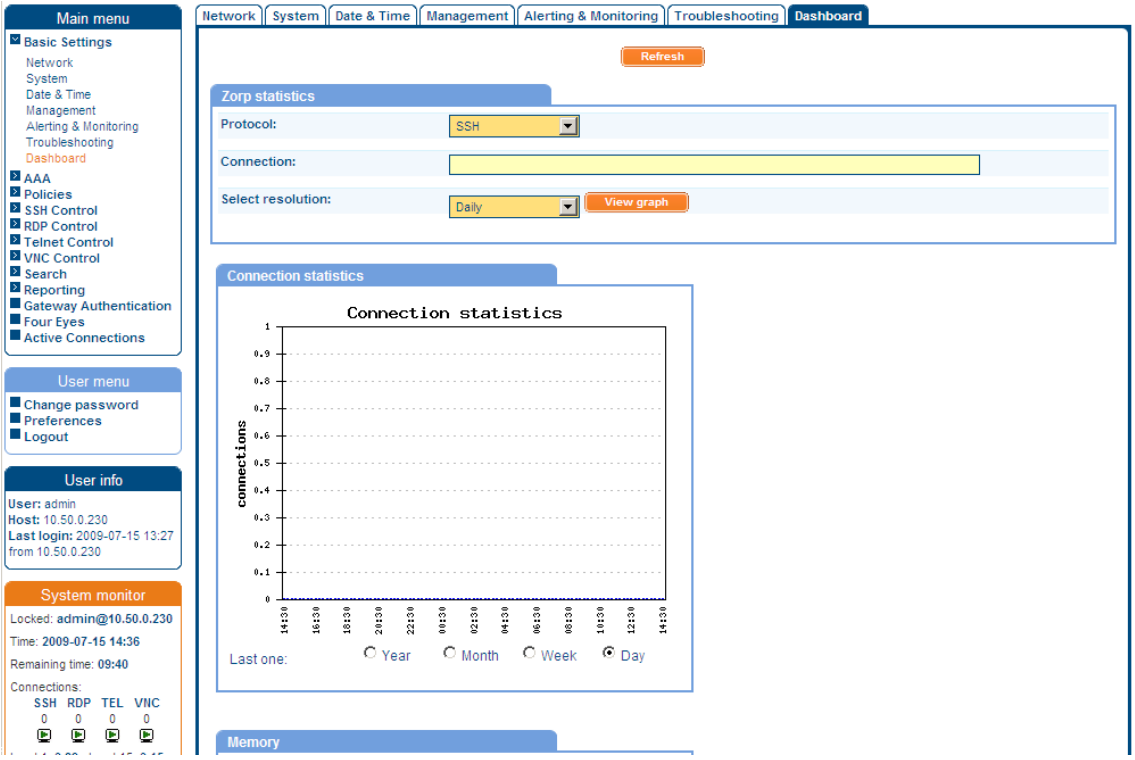


Figure 11.4. The dashboard

The following modules are displayed on the dashboard of SCB:

- *Connection statistics*: Number of active connections per protocol.
- *Memory*: The memory used by the system.
- *Disk*: Filesystem usage for the different partitions.
- *CPU*: CPU usage.
- *Network connections*: Number of network connections.
- *External interface*: Traffic on the external interface.
- *Internal interface*: Traffic on the internal interface.
- *Management interface*: Traffic on the management interface.
- *Load average*: Average load of the system.
- *Processes*: The number of running processes.

11.6.1. Procedure – Displaying custom connection statistics

Purpose:

To display statistics of a specific connection policy, complete the following procedure:

Steps:

Step 1. Navigate to **Basic Settings > Dashboard > Connection statistics**.

Step 2. To display the statistics of a connection policy, enter the name of the policy into the **Connection**.

Step 3. Select the time period to display from the **Select resolution** field.

Step 4. Click **View graph**.

11.7. Troubleshooting an SCB cluster

11.7. Troubleshooting an SCB cluster

The following sections help you to solve problems related to high availability clusters.

- For details on how to recover a cluster that has broken down, see *Procedure 11.7.1, Recovering SCB if both nodes broke down (p. 242)*.
- For details on how to resolve a split-bran situation when the nodes of the cluster were simultaneously active for a time, see *Procedure 11.7.2, Recovering from a split brain situation (p. 243)*.
- For details on replacing a broken node with a new appliance, see *Procedure 11.7.3, Replacing a HA node (p. 245)*.

11.7.1. Procedure – Recovering SCB if both nodes broke down

Purpose:

It can happen that both nodes break down simultaneously (for example because of a power failure), or the slave node breaks down before the original master node recovers. To properly recover SCB, complete the following steps:

Note
As of SCB version 2.0.2, when both nodes of a cluster boot up in parallel, the node with the `1.2.4.1` HA IP address will become the master node.

Steps:

Step 1. Power off both nodes by pressing and releasing the power button.

Warning
Hazard of data loss! If SCB does not shut off, press and hold the power button for approximately 4 seconds. This method terminates connections passing SCB and might result in data loss.

Step 2. Power on the node that was the master before SCB broke down. Consult the system logs to find out which node was the master before the incident: when a node boots as master, or when a takeover occurs, SCB sends a log message identifying the master node.

Tip
Configure remote logging to send the log messages of SCB to a remote server where the messages are available even if the logs stored on SCB become unaccessible. For details on configuring remote logging, see *Section 4.3.3, System logging, SNMP and e-mail alerts (p. 52)*.

Step 3. Wait until this node finishes the boot process.

Step 4. Power on the other node.

11.7.2. Procedure – Recovering from a split brain situation

Purpose:

A split brain situation is caused by a temporary failure of the network link between the cluster nodes, resulting in both nodes switching to the active (master) role while disconnected. This might cause that new data (for example audit trails) is created on both nodes without being replicated to the other node. Thus, it is likely in this situation that two diverging sets of data have been created, which cannot be trivially merged.

Warning
Hazard of data loss! In a split brain situation, valuable audit trails might be available on both SCB nodes, so special care must be taken to avoid data loss.

The nodes of the SCB cluster automatically recognize the split brain situation once the connection between the nodes is reestablished, and do not perform any data synchronization to prevent data loss. When a split brain situation is detected, it is visible on the SCB system monitor, in the system logs (`Split-Brain detected, dropping connection!`), and SCB sends an alert as well.

To recover an SCB cluster from a split brain situation, complete the following steps.

Warning
Do NOT shut down the nodes.

Steps:

Step 1. Temporarily disable all traffic going through SCB. Navigate to **Basic Settings > System > System control** and click **Stop** in the **All traffic** field.
If the web interface is not accessible or unstable, complete the following steps:

Step a. Login to SCB as `root` locally (or remotely using SSH) to access the Console menu.

Step b. Select **Shells > Core Shell**, and issue the `zorpctl stop` command.

Step c. Issue the `date` and check the system date and time. If it is incorrect (for example it displays 2000 January), replace the system battery. For details, see the hardware manual of the appliance.

Step d. Repeat the above steps on the other SCB node.

Step 2. *Optional step for data recovery*: Check the audit trails saved on the SCB nodes.

Step a. Login to the node from a local console.

Step b. Select **Shells > Core Shell** and enter `cd /var/lib/zorp/audit`. The audit trails are located under this directory.

Step c. Find which files were modified since the split brain situation occurred. Use the `find . -mtime -n"` to find the files modified during the last *n*24* hours, or the `find . -mmin -n` to find the files modified during the last *n* minutes.

Step 3. Decide which node should be the master node from now on, then perform the following steps on the to-be-slave node:

Step a. Login to the node from a local console.

Step b. *Optional step for data recovery*: Select **Shells > Core shell** and enter `cd /var/lib/zorp/audit`. The audit trails are located under this directory.

Step c. *Optional step for data recovery*: Backup the audit trails that were modified since the split brain situation occurred.

Warning
This data will be deleted from the SCB node when the split-brain situation is resolved There is no way to import this data back into the database of SCB; it will be available only for offline use.

Step d. *Optional step for data recovery*: To save the corresponding information that can be seen on the **Search** page, export the connection database using the `su postgres -c 'pg_dump scb -f /tmp/database.sql'` command, then backup the `/tmp/database.sql` file.

Step e. *Optional step for data recovery*: Type `exit` to return to the console menu.

Step f. Select **Shells > Boot shell**. If the to-be-slave node is not already the slave node, fail over the cluster to the other node manually by issuing the `/usr/share/heartbeat/hb_standby` command.

Step g. Stop the core firmware. Issue the `/etc/init.d/boot-xcb stop` command.

Step h. Invalidate the DRBD. Issue the following commands:

```
/sbin/drbdsetup /dev/drbd0 disconnect

/sbin/drbdsetup /dev/drbd0 invalidate.
```

Step 4. Reboot the to-be-slave node.

Step 5. Reboot the to-be-master node. The SCB cluster will be now functional, accepting traffic as before.

Step 6. After both nodes reboot, the cluster should be in **Degraded Sync** state, the master being **SyncSource** and the slave being **SyncTarget**. The master node should start synchronizing its data to the slave node. Depending on the amount of data, this can take a long time. To adjust the speed of the synchronization, see *Section 4.5.2.2, Adjusting the synchronization speed of DRBD (p. 97)*.

11.7.3. Procedure – Replacing a HA node

Purpose:

To replace a unit in an SCB cluster with a new appliance, complete the following steps.

Steps:

Step 1. Verify the HA status on the working node. Select **Basic Settings > High Availability**. If one of the nodes has broken down or is missing, the **Status** field displays `DEGRADED`.

Step 2. Perform a full system backup. Before replacing the node, create a complete system backup of the working node. For details, see *Section 4.3.5, Data and configuration archiving and backups (p. 62)*.

Step 3. Check which firmware version is running on the working node. Select **Basic Settings > System > Version details** and write down the exact version numbers.

Step 4. Login to your MyBalaBit account at *https://www.balabit.com/mybalabit/* and download the CD ISO for the same SCB version that is running on your working node.

Step 5. Without connecting the replacement unit to the network, install the replacement unit from the ISO file. Use the IPMI interface if needed.

Step 6. When the installation is finished, connect the two SCB units with an Ethernet cable via the Ethernet connectors labeled as `HA`.

Step 7. Reboot the replacement unit and wait until it finishes booting.

Step 8. Login to the working node and verify the HA state. Select **Basic Settings > High Availability**. The **Status** field should display `HALF`.

Step 9. Click **Other node > Join HA**.

Step 10. Click **Other node > Reboot**.

Step 11. The replacement unit will reboot and start synchronizing data from the working node. The **Basic Settings > High Availability > Status** field will display `DEGRADED SYNC` until the synchronization finishes. Depending on the size of the hard disks and the amount of data stored, this can take several hours.

Step 12. After the synchronization is finished, connect the other Ethernet cables (`EXT`, `INT`, `MGMT`) as needed for your environment.
Expected result:

A node of the SCB cluster is replaced with a new appliance.

11.8. Procedure – Restoring configuration from a configuration backup

Purpose:

In case of a system failure, you can restore your configuration from an earlier configuration backup. To do this, perform the following steps.

Prerequisites:

Before restoring configuration from a configuration backup, a previously existing configuration backup is required.

- Create a configuration backup. For details, see *Procedure 4.3.5.1, Creating configuration and data backups (p. 62)*.

- Optionally, encrypt your configuration backup with GPG. For details, see *Procedure 4.3.5.3, Encrypting configuration backups with GPG (p. 69)*. The following procedure describes the steps for the restoraton of a GPG-encrypted configuration backup.

Steps:

Step 1. Decrypt the configuration with your own private key.

Note
It is not possible to directly import a GPG-encrypted configuration into SCB, it has to be decrypted locally first.

Note
The system backup restoration (**Basic Settings > Management > System backup > Restore now**) only restores statistics, reports, and so on, but does not restore the configuration.

Step 2. If the web interface is still functional, upload the decrypted configuration.

Step a. Navigate to **Basic Settings > System > Import configuration**.

Step b. **Browse** for the configuration file, enter the **Decryption password** if there is any, and click **Upload**.

Step 3. If the web interface is not functional, reinstall SCB and then import the configuration file in the Welcome Wizard. For details, see *Procedure 3.2, Configuring SCB with the Welcome Wizard (p. 29)*.

Appendix 1. About the Secure Shell protocol in a nutshell

The original version of the protocol (SSH-1, dated 1995) has been revised in 1996, and SSH-2 was created offering improved security and new features. The two versions of the protocol are incompatible with each other. Since SSH-1 has inherent design flaws and is vulnerable to attacks, it is now generally considered obsolete and its use should not be permitted. Practically all servers and client application today support SSH-2, use of SSH-1 should be explicitly disabled. Regrettably, software not supporting SSH-2 may still be in use by organizations and it means a considerable security vulnerability to these organizations.

SSH is the main tool used by system administrators for remote system administration tasks. There are several different implementations of the SSH protocol virtually for all kinds of platforms (Windows, Linux, Unix, Mac OS X, BSD, and so on), starting from embedded systems through desktops up to mainframes. Implementations differ in their availability (open source, freeware, commercial) and the level of completeness they offer: some applications implement only certain parts of the protocol, for example WinSCP provides only secure file transfer.

1.1. The basic operation of SSH

One of the main features of the SSH protocol is that almost the entire communication between the client and the server is encrypted – including the authentication of the user. (Naturally, the negotiation of the encryption method to be used is in plain text). During the initialization of the session server authentication is performed and parameters for encryption, data compression and integrity verification of the data transferred are negotiated. The protocol enforces user authentication and is capable of authenticating the user via various methods: password, RSA key, Challenge/Response schemes like S/Key and OPIE, and so on The typical uses of SSH include the following:

- Remote shell: Remotely administer a computer via an interactive terminal console. This is one of the most widespread uses of SSH.
- Remote command execution: Execute commands on the remote machine. Remote command execution can result in significant data transfer, for example when performing scheduled or manual tasks such as file copying (scp), data or file synchronization (rsync), creating archive backups (tar), and so on.
- TCP IP forwarding (also known as port forwarding): It is possible to tunnel any TCP/IP connection from the client or from the server into the encrypted SSH channel. It can also be used to forward communication otherwise not allowed, such as the access of ports banned by the security policy. This allows to secure any – normally unencrypted – data transfer and is frequently used as an easy way to secure connections between the hosts without the need to set up full VPN connections.
- File transfer: Securely transfer files using SFTP.
- X11 forwarding: Applications running on the server and requiring graphical interface (X Window) appear on the client's monitor, but run on the server in all other respect, thus it is possible to work with them remotely.
- Agent forwarding: Transfer authentication requests to the client machine.

Depending on the server and client application, SSH is able to transfer data of several different channels simultaneously within a single connection, thus for example it is possible to transfer files from the client to the server via SFTP and to manage the server remotely via remote shell using only a single connection.

1.2. Configuring encryption parameters

SCB is able to enforce policies on the various elements of the encrypted SSH communication, such as the MAC, key-exchange, and so on algorithms that are permitted to be used. The parameters can be set separately for the client and for the server side. The attributes are represented as comma-separated strings listing the enabled methods/algorithms, in the order of preference.

The following parameters can be configured:

- *Key exchange (KEX) algorithms*: SCB supports the `diffie-hellman-group14-sha1` and `diffie-hellman-group1-sha1` algorithms.
- *Hostkey algorithms*: The supported algorithms are `ssh-rsa` and `ssh-dss`.

Note
To show the clients a hostkey using the above algorithms, the corresponding private key has to be set in the connection policy of SCB.

- *Cipher algorithms*: The following symmetric-cipher algorithms are supported: `aes128-cbc`, `3des-cbc`, `blowfish-cbc`, `cast128-cbc`, `arcfour`, `aes192-cbc`, `aes256-cbc`.
- *MAC algorithms*: The supported algorithms are: `hmac-sha1` and `hmac-md5`.

Appendix 2. Package contents inventory

Carefully unpack all server components from the packing cartons. The following items should be packaged with the BalaBit Shell Control Box:

- A BalaBit Shell Control Box appliance, preinstalled with the latest BalaBit Shell Control Box firmware.
- BalaBit Shell Control Box accessory kit, including the following:
  - Delivery note
  - BalaBit Shell Control Box License Agreement
  - BalaBit Shell Control Box Certificate (includes the purchased license and support options, BIOS and IPMI passwords, and support contact details)
  - BalaBit Shell Control Box Hardware Installation Guide
- Rackmount
- Power cable

Appendix 3. BalaBit Shell Control Box Hardware Installation Guide

Appendix 3. BalaBit Shell Control Box Hardware Installation Guide

This leaflet describes how to set up the BalaBit Shell Control Box (SCB) hardware. Refer to the following documents for step-by-step instructions:

- *BalaBit Shell Control Box N1000*: For details on installing SCB into a rack, see the *SC811 CHASSIS Series User's Manual, Chapter 6: Rack Installation*. For details on connecting the cables to SCB see the *X8SIE/X8SIE-F/X8SIE-LN4/X8SI6-F User's Manual, Section 2-5 Connectors/IO Ports*.
- *BalaBit Shell Control Box N1000d*: For details on installing SCB into a rack, see the *SC825M CHASSIS Series User's Manual, Chapter 5: Rack Installation*. For details on connecting the cables to SCB see the *X8DT3/X8DTi/X8DT3-F/X8DTi-F/X8DT3-LN4F/X8DTi-LN4F User's Manual, Section 2-5 Control Panel Connectors/IO Ports*.
- *BalaBit Shell Control Box N10000*: For details on installing SCB into a rack, see the *SC826 CHASSIS Series User's Manual, Chapter 6: Rack Installation*. For details on connecting the cables to SCB see the *X8DT3/X8DTi/X8DT3-F/X8DTi-F/X8DT3-LN4F/X8DTi-LN4F User's Manual, Section 2-5 Control Panel Connectors/IO Ports*.

The manuals are available online at *the BalaBit Hardware Documentation page*.

- For details on how to install a single SCB unit, see *Procedure 3.1, Installing the SCB hardware (p. 250)*.
- For details on how to install a two SCB units in high availability mode, see *Procedure 3.2, Installing two SCB units in HA mode (p. 251)*.

3.1. Procedure – Installing the SCB hardware

Purpose:

To install a single SCB unit, complete the following steps.

Steps:

Step 1. Unpack SCB.

Step 2. *Optional step*: Install SCB into a rack with the slide rails. Slide rails are available for all SCB appliances.

Step 3. Connect the cables.

Step a. Connect the Ethernet cable facing your LAN to the Ethernet connector labeled as `EXT`. This is the external interface of SCB and is used to configure SCB. (For details on the roles of the different interfaces, see *Section 2.9, Network interfaces (p. 18)*.)

Step b. Connect the Ethernet cable facing your servers to the Ethernet connector labeled as `INT`. This is the internal interface of SCB. (For details on the roles of the different interfaces, see *Section 2.9, Network interfaces (p. 18)*.)

Step c. Connect an Ethernet cable that you can use to remotely support the SCB hardware to the `IPMI` interface of SCB. For details, see the following documents:

- The *Onboard BMC/IPMI User's Guide*, available at *the BalaBit Hardware Documentation page*.

It is not necessary for the IPMI interface to be accessible from the Internet, but the administrator of SCB must be able to access it for support and troubleshooting purposes in case vendor support is needed.

Warning
Access to information available only via the IPMI interface is a not mandatory, but highly recommended to speed up the support and troubleshooting processes.

Step d. *Optional step*: Connect the Ethernet cable to be used for managing SCB after its initial configuration to the Ethernet connector labeled as `MGMT`. This is the management interface of SCB. (For details on the roles of the different interfaces, see *Section 2.9, Network interfaces (p. 18)*.)

Step e. *Optional step*: Connect the Ethernet cable connecting SCB to another SCB node to the Ethernet connector labeled as `HA`. This is the high availability (HA) interface of SCB. (For details on the roles of the different interfaces, see *Section 2.9, Network interfaces (p. 18)*.)

Step 4. Power on the hardware.

Step 5. Change the BIOS and IPMI passwords on the BalaBit Shell Control Box. The original passwords are printed on the BalaBit Shell Control Box Certificate received from BalaBit.

Step 6. Connect to the SCB web interface from a client machine and complete the Welcome Wizard. This might require you to configure an alias interface on the client machine. Step 5 is described in detail in *Chapter 3, The Welcome Wizard and the first login (p. 23)*.

Note
The BalaBit Shell Control Box Administrator Guide is available on the SCB on the *BalaBit* Documentation page.

3.2. Procedure – Installing two SCB units in HA mode

Purpose:

To install SCB with high availability support, complete the following steps.

Steps:

Step 1. For the first SCB unit, complete *Procedure 3.1, Installing the SCB hardware (p. 250)*.

Step 2. For the second SCB unit, complete Steps 1-2 of *Procedure 3.1, Installing the SCB hardware (p. 250)*.

Step 3. Connect the two units with an Ethernet cable via the Ethernet connectors labeled as `HA`.

Step 4. Power on the second unit.

Step 5. Connect to the SCB web interface of the first unit from a client machine and enable the high availability mode. Navigate to **Basic Settings > High Availability** . Click **Convert to Cluster**, then reload the page in your browser.

Step 6. Reboot the slave unit, then reboot the master unit. Wait a few minutes until the master unit boots, then reload the page in your browser.

Step 7. Wait until the slave unit synchronizes its disk to the master unit. Depending on the size of the hard disks, this may take several hours. You can increase the speed of the synchronization via the SCB web interface at **Basic Settings > High Availability > DRBD sync rate limit**.

Refer to the *Setup Troubleshooting* chapter of the respective guide if you encounter any problems. If you still experience problems, contact the BalaBit Support Team via phone or e-mail:

Support e-mail address: <support@balabit.com>.

Support hotline: +36 1 371 0540 (available from 9 AM to 5 PM CET on weekdays)

If you have already registered your product on the *MyBalaBit* website, submit your support request using the *BalaBit Online Support System*. This online support service is available 24 hours a day.

Appendix 4. BalaBit Shell Control Box VMware Installation Guide

This tutorial describes the possibilities and limitations of installing BalaBit Shell Control Box (SCB) 3 LTS as a virtual appliance under a VMware ESXi server.

4.1. Limitations of SCB under VMware

Version 3 LTS of SCB has no special support for running under VMware. While the basic functionality of SCB is not affected by running as a virtual appliance, the following limitations apply:

- SCB can use only the fixed disk space assigned to the virtual host during its installation: it is not possible to modify the size of the disks later, neither can SCB be used in on-demand disk allocation scenarios.
- SCB currently does not comply with the VMware Ready program and does not have vmware-tools installed. As a result, SCB does not support VMotion, nor any other VMware features that require vmware-tools. Consult the documentation of your VMware server to check which features work without vmware-tools.
- High availability mode is not supported in VMware.
- Hardware-related alerts and status indicators of SCB may display inaccurate information, for example, display degraded RAID status.
- The Bridge operation mode of SCB cannot be used in VMware.

4.2. Procedure – Installing SCB under VMware ESXi

Purpose:

To install a new SCB under VMware ESXi, complete the following steps:

Warning
SCB can be installed under VMware ESXi 4.0 or later, earlier VMware versions are not supported.

Steps:

Step 1. Create the virtual machine for SCB using the following settings:

- Guest operating system: **Linux/Ubuntu 32bit**
- Allocate memory for the virtual machine. SCB requires a minimum of 512MB of memory. The recommended size for the memory depends on the exact environment, but consider the following:
  - The base system requires 256MB

  - SCB requires about 1-5MB of memory for every active connection, depending on the type of the connection — graphical protocols require more memory.
  - Using more than 8GB of memory is not supported under VMware, with 4GB being adequate for most tasks.
- The hard disk controller must be **LSI Logic Parallel**.
- Do not use RAID for the hard disk, a single hard disk is sufficient for the system.
- SCB uses a single fixed-sized disk. About 5GB is required for the base system, the remaining disk space is used to store data. Use one of the following disk sizes (in GB): *8, 12, 68, 160, 230, 250, 465, 500, 925, 1029, 2053, 3077, 4101, 4645, 5125, 6149, 7173, 8197, 9205*.
- SCB requires 4 network cards, all of them must be **E1000**.

Step 2. After creating the virtual machine, edit the settings of the machine. Set the following options:

Step a. Under **Options > VMware Tools** enable the **Shutdown, Suspend, Reset** options, otherwise the SCB administrator will not be able to access these functions from the SCB web interface.

Step b. Under **Options > Boot options** enable the **Force BIOS Setup** option. This is required to be able to check the system time (and modify it if needed) before installing SCB.

Step c. Under **Options > Paravirtualization** enable the **Support VMI Paravirtualization** option.

Step 3. Login to your *MyBalaBit account* and download the latest BalaBit Shell Control Box installation ISO file. Note that you need to have purchased SCB as a virtual appliance or have partner access to download BalaBit Shell Control Box ISO files. If you are a partner but do not see the ISO files, you can request partner access within MyBalaBit.

Step 4. Mount the ISO image and boot the virtual machine. Follow the on-screen instructions to install SCB.

Appendix 5. Configuring external devices

5.1. Procedure – Configuring advanced routing on Linux

Purpose:

To configure a Linux-based router to redirect selected traffic to SCB instead of its original destination, complete the following steps. This procedure should work on most modern Linux-based routers, including Check Point® routers.

Prerequisites:

The router must have the *iptables* and *ip* tools installed.

Steps:

Step 1. Create the packet filter rules that will mark the connections to be sent to SCB using the CONNMARK feature of iptables. Mark only connections that must be redirected to SCB.

```
# iptables -t mangle -I PREROUTING -i <interface-facing-the-clients> -p tcp
 -d <network-of-the-servers> --dport <port-to-access> -j CONNMARK --set-mark
 1
```

For example, ifthe network interface of the router that faces the clients is called *client*, the servers are located in the *10.0.0.0/24* subnet, and the clients access the servers using port *3389* (the default port of the RPD protocol), then this command looks like: `# iptables -t mangle -I PREROUTING -i client -p tcp -d 10.0.0.0/24 --dport 3389 -j CONNMARK --set-mark 1`

Step 2. Create a rule that redirects the answers of the servers to SCB. That way both the client-to-server and the server-to-client traffic is routed to SCB.

```
# iptables -t mangle -I PREROUTING -i <interface-facing-the-servers> -p tcp
 -s <network-of-the-servers> --sport <port-to-access> -j CONNMARK --set-mark
 1
```

Step 3. Convert the CONNMARK marks to MARK:

```
# iptables -t mangle -A PREROUTING ! -i bond1.1001 -m connmark --mark 1 -j
MARK --set-mark 1
```

Warning
This rule must be placed after the CONNMARK rules.

Step 4. Create a routing table which has a single entry with a default route to SCB:

```
# /sbin/ip route add default via <ip-address-of-SCB> table scb
```

Step 5. Add the table name to the `/etc/iproute2/rt_tables` of the router. Use the following format (for details on routing tables, see for example the *Guide to IP Layer Network Administration with Linux*):

```
103 scb
```

Step 6. Create a routing rule which selects the routing table called *scb* if the connection is marked.

```
# /sbin/ip rule add from all fwmark 1 table scb
```

Step 7. If SCB is configured to spoof the IP address of the clients on the server side (that is, the **SNAT > Use the original IP address of the client** option of the connection policies is selected), enable spoofing on the router for the interface connected to SCB.

```
# echo 0 > /proc/sys/net/ipv4/conf/<interface-facing-SCB>/rp_filter
# echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
```

Expected result:

The traffic the clients targeting the specified port of the servers is redirected to SCB. That way SCB can be configured to configure and audit this traffic.

Appendix 6. BalaBit Shell Control Box End User License Agreement

(c) BalaBit IT Security Ltd.

6.1. SUBJECT OF THE LICENSE CONTRACT

This License Contract is entered into by and between BalaBit and Licensee and sets out the terms and conditions under which Licensee and/or Licensee's Authorized Subsidiaries may use the BalaBit Shell Control Box under this License Contract.

6.2. DEFINITIONS

In this License Contract, the following words shall have the following meanings:

Company name: **BalaBit IT Security Ltd.**

Registered office: H-1115 Budapest, Bártfai u. 54.

Company registration number: 01-09-687127

Tax number: HU11996468

6.3. WORDS AND EXPRESSIONS

Annexed Software	Any third party software that is a not a BalaBit Product contained in the install media of the BalaBit Product.
Authorized Subsidiary	Any subsidiary organization: (i) in which Licensee possesses more than fifty percent (50%) of the voting power and (ii) which is located within the Territory.
BalaBit Product	Any software, hardware or service Licensed, sold, or provided by BalaBit including any installation, education, support and warranty services, with the exception of the Annexed Software.
License Contract	The present BalaBit Shell Control Box License Contract.
Product Documentation	Any documentation referring to the BalaBit Shell Control Box or any module thereof, with special regard to the administration guide, the product description, the installation guide, user guides and manuals.
Protected Servers	Host computers located in the zones protected by BalaBit Shell Control Box that have their administrative traffic controlled or audited by BalaBit Shell Control Box.
Protected Objects	The entire BalaBit Shell Control Box software including all of its modules, all the related Product Documentation; the source code, the structure of the databases, all registered information reflecting the structure of the BalaBit Shell Control Box and all the adaptation and copies of the Protected Objects that presently exist or that are to be developed in the future, or any product falling under the copyright of BalaBit.

BalaBit Shell Control Box	The BalaBit Product designed for controlling and auditing server administration traffic as defined by the Product Description.
Warranty Period	The period of twelve (12) months from the date of delivery of the BalaBit Shell Control Box to Licensee.
Territory	The countries or areas specified above in respect of which Licensee shall be entitled to install and/or use BalaBit Shell Control Box.
End-user Certificate	The document signed by Licensor which contains a) identification data of Licensee; b) configuration of BalaBit Shell Control Box, number of Protected Servers and designation of Licensed modules thereof; c) designation of the Territory; d) declaration of the parties on accepting the terms and conditions of this License Contract; and e) declaration of Licensee that is in receipt of the install media and the hardware appliance.

Table 6.1. Words and expressions

6.4. LICENSE GRANTS AND RESTRICTIONS

For the BalaBit Shell Control Box licensed under this License Contract, BalaBit grants to Licensee a non-exclusive, non-transferable, perpetual license to use such BalaBit Product under the terms and conditions of this License Contract and the applicable End-user Certificate.

Licensee shall use the BalaBit Shell Control Box in the in the configuration and in the quantities specified in the End-user Certificate within the Territory.

On the install media (firmware CD-ROM) all modules of the BalaBit Shell Control Box will be presented, however, Licensee shall not be entitled to use any module which was not Licensed to it. Access rights to modules and IP connections are controlled by an "electronic key" accompanying the BalaBit Shell Control Box.

Licensee shall be entitled to make one back-up copy of the install media containing the BalaBit Shell Control Box.

Licensee shall make available the Protected Objects at its disposal solely to its own employees and those of the Authorized Subsidiaries.

Licensee shall take all reasonable steps to protect BalaBit's rights with respect to the Protected Objects with special regard and care to protecting it from any unauthorized access.

Licensee shall, in 5 working days, properly answer the queries of BalaBit referring to the actual usage conditions of the BalaBit Shell Control Box that may differ or allegedly differs from the License conditions.

Licensee shall not modify the BalaBit Shell Control Box in any way, with special regard to the functions inspecting the usage of the software. Licensee shall install the code permitting the usage of the BalaBit Shell Control Box according to the provisions defined for it by BalaBit. Licensee may not modify or cancel such codes. Configuration settings of the BalaBit Shell Control Box in accordance with the possibilities offered by the system shall not be construed as modification of the software.

Licensee shall only be entitled to analyze the structure of the BalaBit Products (decompilation or reverse- engineering) if concurrent operation with a software developed by a third party is necessary, and upon request to supply the information required for concurrent operation BalaBit does not provide such information within 60 days from the receipt of such a request.

These user actions are limited to parts of the BalaBit Product which are necessary for concurrent operation. *Any information obtained as a result of applying the previous Section (i) cannot be used for purposes other than concurrent operation with the BalaBit Product; (ii) cannot be disclosed to third parties unless it is necessary for concurrent operation with the BalaBit Product; (iii) cannot be used for the development, production or distribution of a different software which is similar to the BalaBit Product in its form of expression, or for any other act violating copyright.*

For any Annexed Software contained by the same install media as the BalaBit Product, the terms and conditions defined by its copyright owner shall be properly applied. BalaBit does not grant any License rights to any Annexed Software.

Any usage of the BalaBit Shell Control Box exceeding the limits and restrictions defined in this License Contract shall qualify as material breach of the License Contract.

The Number of Protected Servers shall not exceed the amount defined in the End-user Certificate.

Licensee shall have the right to obtain and use content updates only if Licensee concludes a maintenance contract that includes such content updates, or if Licensee has otherwise separately acquired the right to obtain and use such content updates. This License Contract does not otherwise permit Licensee to obtain and use content updates.

6.5. SUBSIDIARIES

Authorized Subsidiaries may also utilize the services of the BalaBit Shell Control Box under the terms and conditions of this License Contract. Any Authorized Subsidiary utilizing any service of the BalaBit Shell Control Box will be deemed to have accepted the terms and conditions of this License Contract.

6.6. INTELLECTUAL PROPERTY RIGHTS

Licensee agrees that BalaBit owns all rights, titles, and interests related to the BalaBit Shell Control Box and all of BalaBit's patents, trademarks, trade names, inventions, copyrights, know-how, and trade secrets relating to the design, manufacture, operation or service of the BalaBit Products.

The use by Licensee of any of these intellectual property rights is authorized only for the purposes set forth herein, and upon termination of this License Contract for any reason, such authorization shall cease.

The BalaBit Products are Licensed only for internal business purposes in every case, under the condition that such License does not convey any license, expressly or by implication, to manufacture, duplicate or otherwise copy or reproduce any of the BalaBit Products. No other rights than expressly stated herein are granted to Licensee.

Licensee will take appropriate steps with its Authorized Subsidiaries, as BalaBit may request, to inform them of and assure compliance with the restrictions contained in the License Contract.

6.7. TRADE MARKS

BalaBit hereby grants to Licensee the non-exclusive right to use the trade marks of the BalaBit Products in the Territory in accordance with the terms and for the duration of this License Contract.

BalaBit makes no representation or warranty as to the validity or enforceability of the trade marks, nor as to whether these infringe any intellectual property rights of third parties in the Territory.

6.8. NEGLIGENT INFRINGEMENT

In case of negligent infringement of BalaBit's rights with respect to the BalaBit Shell Control Box, committed by violating the restrictions and limitations defined by this License Contract, Licensee shall pay liquidated damages to BalaBit. The amount of the liquidated damages shall be twice as much as the price of the BalaBit Product concerned, on BalaBit's current Price List.

6.9. INTELLECTUAL PROPERTY INDEMNIFICATION

BalaBit shall pay all damages, costs and reasonable attorney's fees awarded against Licensee in connection with any claim brought against Licensee to the extent that such claim is based on a claim that Licensee's authorized use of the BalaBit Product infringes a patent, copyright, trademark or trade secret. Licensee shall notify BalaBit in writing of any such claim as soon as Licensee learns of it and shall cooperate fully with BalaBit in connection with the defense of that claim. BalaBit shall have sole control of that defense (including without limitation the right to settle the claim).

If Licensee is prohibited from using any BalaBit Product due to an infringement claim, or if BalaBit believes that any BalaBit Product is likely to become the subject of an infringement claim, BalaBit shall at its sole option, either: (i) obtain the right for Licensee to continue to use such BalaBit Product, (ii) replace or modify the BalaBit Product so as to make such BalaBit Product non-infringing and substantially comparable in functionality or (iii) refund to Licensee the amount paid for such infringing BalaBit Product and provide a pro-rated refund of any unused, prepaid maintenance fees paid by Licensee, in exchange for Licensee's return of such BalaBit Product to BalaBit.

Notwithstanding the above, BalaBit will have no liability for any infringement claim to the extent that it is based upon: (i) modification of the BalaBit Product other than by BalaBit, (ii) use of the BalaBit Product in combination with any product not specifically authorized by BalaBit to be combined with the BalaBit Product or (iii) use of the BalaBit Product in an unauthorized manner for which it was not designed.

6.10. LICENSE FEE

The number of the Protected Servers, the configuration and the modules licensed shall serve as the calculation base of the License fee.

Licensee acknowledges that payment of the License fees is a condition of lawful usage.

License fees do not contain any installation or post charges.

6.11. WARRANTIES

BalaBit warrants that during the Warranty Period, the magnetic or optical media upon which the BalaBit Product is recorded will not be defective under normal use. BalaBit will replace any defective media returned to it, accompanied by a dated proof of purchase, within the Warranty Period at no charge to Licensee. Upon receipt of the allegedly defective BalaBit Product, BalaBit will at its option, deliver a replacement BalaBit Product or BalaBit's current equivalent to Licensee at no additional cost. BalaBit will bear the delivery charges to Licensee for the replacement Product.

In case of installation by BalaBit, BalaBit warrants that during the Warranty Period, the BalaBit Shell Control Box, under normal use in the operating environment defined by BalaBit, and without unauthorized modification, will perform in substantial compliance with the Product Documentation accompanying the BalaBit Product, when used

on that hardware for which it was installed, in compliance with the provisions of the user manuals and the recommendations of BalaBit. The date of the notification sent to BalaBit shall qualify as the date of the failure. Licensee shall do its best to mitigate the consequences of that failure. If, during the Warranty Period, the BalaBit Product fails to comply with this warranty, and such failure is reported by Licensee to BalaBit within the Warranty Period, BalaBit's sole obligation and liability for breach of this warranty is, at BalaBit's sole option, either: (i) to correct such failure, (ii) to replace the defective BalaBit Product or (iii) to refund the license fees paid by Licensee for the applicable BalaBit Product.

6.12. DISCLAIMER OF WARRANTIES

EXCEPT AS SET OUT IN THIS LICENSE CONTRACT, BALABIT MAKES NO WARRANTIES OF ANY KIND WITH RESPECT TO THE BALABIT SHELL CONTROL BOX. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, BALABIT EXCLUDES ANY OTHER WARRANTIES, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF SATISFACTORY QUALITY, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS.

6.13. LIMITATION OF LIABILITY

SOME STATES AND COUNTRIES, INCLUDING MEMBER COUNTRIES OF THE EUROPEAN UNION, DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES AND, THEREFORE, THE FOLLOWING LIMITATION OR EXCLUSION MAY NOT APPLY TO THIS LICENSE CONTRACT IN THOSE STATES AND COUNTRIES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW AND REGARDLESS OF WHETHER ANY REMEDY SET OUT IN THIS LICENSE CONTRACT FAILS OF ITS ESSENTIAL PURPOSE, IN NO EVENT SHALL BALABIT BE LIABLE TO LICENSEE FOR ANY SPECIAL, CONSEQUENTIAL, INDIRECT OR SIMILAR DAMAGES OR LOST PROFITS OR LOST DATA ARISING OUT OF THE USE OR INABILITY TO USE THE BALABIT SHELL CONTROL BOX EVEN IF BALABIT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IN NO CASE SHALL BALABIT'S TOTAL LIABILITY UNDER THIS LICENSE CONTRACT EXCEED THE FEES PAID BY LICENSEE FOR THE BALABIT SHELL CONTROL BOX LICENSED UNDER THIS LICENSE CONTRACT.

6.14. DURATION AND TERMINATION

This License Contract shall come into effect on the date of signature of the End-user Certificate by the duly authorized representative of BalaBit.

Licensee may terminate the License Contract at any time by written notice sent to BalaBit and by simultaneously destroying all copies of the Protected Objects licensed under this License Contract.

BalaBit may terminate this License Contract with immediate effect by written notice to Licensee, if Licensee is in material or persistent breach of the License Contract and either that breach is incapable of remedy or Licensee shall have failed to remedy that breach within 30 days after receiving written notice requiring it to remedy that breach.

6.15. AMENDMENTS

Save as expressly provided in this License Contract, no amendment or variation of this License Contract shall be effective unless in writing and signed by a duly authorized representative of the parties to it.

6.16. WAIVER

The failure of a party to exercise or enforce any right under this License Contract shall not be deemed to be a waiver of that right nor operate to bar the exercise or enforcement of it at any time or times thereafter.

6.17. SEVERABILITY

If any part of this License Contract becomes invalid, illegal or unenforceable, the parties shall in such an event negotiate in good faith in order to agree on the terms of a mutually satisfactory provision to be substituted for the invalid, illegal or unenforceable provision which as nearly as possible validly gives effect to their intentions as expressed in this License Contract.

6.18. NOTICES

Any notice required to be given pursuant to this License Contract shall be in writing and shall be given by delivering the notice by hand, or by sending the same by prepaid first class post (airmail if to an address outside the country of posting) to the address of the relevant party set out in this License Contract or such other address as either party notifies to the other from time to time. Any notice given according to the above procedure shall be deemed to have been given at the time of delivery (if delivered by hand) and when received (if sent by post).

6.19. MISCELLANEOUS

Headings are for convenience only and shall be ignored in interpreting this License Contract.

This License Contract and the rights granted in this License Contract may not be assigned, sublicensed or otherwise transferred in whole or in part by Licensee without BalaBit's prior written consent. This consent shall not be unreasonably withheld or delayed.

An independent third party auditor, reasonably acceptable to BalaBit and Licensee, may upon reasonable notice to Licensee and during normal business hours, but not more often than once each year, inspect Licensee's relevant records in order to confirm that usage of the BalaBit Shell Control Box complies with the terms and conditions of this License Contract. BalaBit shall bear the costs of such audit. All audits shall be subject to the reasonable safety and security policies and procedures of Licensee.

This License Contract constitutes the entire agreement between the parties with regard to the subject matter hereof.

Any modification of this License Contract must be in writing and signed by both parties.

Appendix 7. Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd) License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED. BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. *Definitions*
   a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
   b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.
   c. "Distribute" means to make available to the public the original and copies of the Work through sale or other transfer of ownership.
   d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
   e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
   f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing;

a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. *Fair Dealing Rights.* Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. *License Grant.* Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

   a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections; and,

   b. to Distribute and Publicly Perform the Work including as incorporated in Collections.

   The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Adaptations. Subject to 8(f), all rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Section 4(d).

4. *Restrictions.* The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

   a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every

copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested.

b. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

c. If You Distribute, or Publicly Perform the Work or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (for example a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Collection, at a minimum such credit will appear, if a credit for all contributing authors of Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

d. For the avoidance of doubt:

   i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

   ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License if Your exercise of such rights is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(b) and otherwise waives the right to collect royalties through any statutory or compulsory licensing scheme; and,

   iii. Voluntary License Schemes. The Licensor reserves the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this Li-

cense that is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(b).

e. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation.

5. *Representations, Warranties and Disclaimer* UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. *Limitation on Liability.* EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. *Termination*

   a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

   b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. *Miscellaneous*

   a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

   b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

   c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

e. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Glossary

4-eyes authorization	4-eyes authorization is an advanced authorization method where only two administrators logging in simultaneously are permitted to access the server. These administrators can monitor each other's work, reducing the chance of (accidental or intentional) human errors in the server administration process.
alias IP	An additional IP address assigned to an interface that already has an IP address. The normal and alias IP addresses both refer to the same physical interface.
auditing policy	The auditing policy determines which events are logged on host running Microsoft Windows operating systems.
authentication	The process of verifying the authenticity of a user or client before allowing access to a network system or service.
Authentication Policy	An authentication policy is a list of authentication methods that can be used in a connection. Connection definitions refer to an authentication policy to determine how the client can authenticate to the target server.
Audit trail	An audit trail is a file storing the recorded activities of the administrators in an encrypted format. Audit trails can be replayed using the Audit Player application.
Audit Player	Audit Player is a desktop application that can replay recorded audit trails like movie. The Audit Player is available for the Microsoft Windows and GNU/Linux platforms.
BSD-syslog protocol	The old syslog protocol standard described in RFC 3164 *The BSD syslog Protocol*. Sometimes also referred to as the legacy-syslog protocol.
CA	A Certificate Authority (CA) is an institute that issues certificates.
certificate	A certificate is a file that uniquely identifies its owner. Certificates contains information identifying the owner of the certificate, a public key itself, the expiration date of the certificate, the name of the CA that signed the certificate, and some other data.
Channel Policy	The channel policy lists the SSH channels (for example terminal session, SCP, and so on) that can be used in a connection. The channel policy can further restrict access to each channel based on the IP address of the client or the server, a user list, or a time policy.
client mode	In client mode, syslog-ng collects the local logs generated by the host and forwards them through a network connection to the central syslog-ng server or to a relay.
controlled traffic	SCB audits and controls only the traffic that is configured in the connection and channel policies, all other traffic is forwarded on the packet level without any inspection.

disk buffer	The Premium Edition of syslog-ng can store messages on the local hard disk if the central log server or the network connection to the server becomes unavailable.
disk queue	See *disk buffer*.
domain name	The name of a network, for example `balabit.com`.
External network interface	The *external* interface (labeled `EXT`) is used for general communication between the clients and the servers. If the SCB appliance has only two interfaces, the external interface is used for management purposes as well.
firmware	A firmware is a collection of the software components running on SCB. Individual software components cannot be upgraded on SCB, only the entire firmware. SCB contains two firmwares, an external (or boot) firmware and an internal (or core) firmware. These can be upgraded separately.
gateway	A device that connect two or more parts of the network, for example your local intranet and the external network (the Internet). Gateways act as entrances into other networks.
High Availability	High Availability (HA) uses a second SCB unit (called slave node) to ensure that the services are available even if the first unit (called master node) breaks down.
host	A computer connected to the network.
hostname	A name that identifies a host on the network. Hostnames can contain only alpha-numerical characters (A-Z, a-z, 0-9) and the hyphen (-) character.
HA network interface	The *HA* interface (labeled `HA`) is an interface reserved for communication between the nodes of SCB clusters.
Internal network interface	The *internal* interface (labeled `INT`) is used exclusively for communication between SCB and the protected servers; incoming connections are not allowed on this interface.
IETF-syslog protocol	The syslog-protocol standard developed by the Internet Engineering Task Force (IETF), described in RFC 5424 *The BSD syslog Protocol*.
key pair	A private key and its related public key. The private key is known only to the owner; the public key can be freely distributed. Information encrypted with the private key can only be decrypted using the public key.
LDAP	The Lightweight Directory Access Protocol (LDAP), is an application protocol for querying and modifying data using directory services running over TCP/IP.
License	SCB's license determines the number of servers (IP addresses) that SCB protects; the license limits the number of IP addresses accessible on the internal interface.

log path	A combination of sources, filters, parsers, rewrite rules, and destinations: syslog-ng examines all messages arriving to the sources of the logpath and sends the messages matching all filters to the defined destinations.
Management network interface	The *management* interface (labeled `MGMT`) is used exclusively for communication between SCB and the auditor or the administrator of the BalaBit Shell Control Box. This includes replaying the audit trails stored on SCB using the Audit Player application (AP).
master node	The active SCB unit that is inspecting the traffic when SCB is used in High Availability mode.
name server	A network computer storing the IP addresses corresponding to domain names.
node	An SCB unit running in High Availability mode.
ping	A command that sends a message from a host to another host over a network to test connectivity and packet loss.
port	A number ranging from 1 to 65535 that identifies the destination application of the transmitted data. For example: SSH commonly uses port 22, web servers (HTTP) use port 80, and so on.
Public-key authentication	An authentication method that uses encryption key pairs to verify the identity of a user or a client.
redundant Heartbeat interface	A redundant Heartbeat interface is a virtual interface that uses an existing interface of the SCB device to detect that the other node of the SCB cluster is still available. The virtual interface is not used to synchronize data between the nodes, only Heartbeat messages are transferred.
SCB	An abbreviation of the BalaBit Shell Control Box name.
slave node	The passive SCB unit that replaces the active unit (the master node) if the master becomes unavailable.
SNMP	Simple Network Management Protocol (SNMP) is an industry standard protocol used for network management. SCB can send SNMP alerts to a central SNMP server.
split brain	A split brain situation occurs when for some reason (for example the loss of connection between the nodes) both nodes of a SCB cluster become active (master). This might cause that new data (for example audit trails) is created on both nodes without being replicated to the other node. Thus, it is likely in this situation that two diverging sets of data are created, which cannot be trivially merged.
SSH settings	SSH settings determine the parameters of the connection on the protocol level, including timeout value and greeting message of the connection, as well as the encryption algorithms used.

syslog-ng	The syslog-ng application is a flexible and highly scalable system logging application, typically used to manage log messages and implement centralized logging.
syslog-ng client	A host running syslog-ng in client mode.
syslog-ng Premium Edition	The syslog-ng Premium Edition is the commercial version of the open-source application. It offers additional features, like encrypted message transfer and an agent for Microsoft Windows platforms.
syslog-ng relay	A host running syslog-ng in relay mode.
syslog-ng server	A host running syslog-ng in server mode, like SCB.
Time Policy	The time policy determines which hours of a day can the users access a connection or a channel.
traceroute	A command that shows all routing steps (the path of a message) between two hosts.
User List	User lists are white- or blacklists of usernames that allow fine-control over who can access a connection or a channel.

Index

Symbols

4-eyes authorization, 14-15, 17, 129, 208, 215

A

accessing SCB using SSH, 111
  in Bastion mode, 224
accounting SCB, 90, 178
admin password, 33
administrator password, 33
advanced routing, 255 (see advanced routing)
agent-forwarding, 152, 158-159
alias interfaces, 47
alias IP addresses, 24, 28, 47
AP, 191
  add new font, 204
  context search in RDP streams, 204
  encrypted audit trails, 202
  finding audit trails, 200
  font types in RDP streams, 204
  hotkeys, 197
  importing certificates, 202
  live mode, 196, 218
  logging in, 205
  OCR, 204
  projects, 202
  replaying audit trails, 194
  saving results, 202
  saving trailsets, 202
  searching in RDP streams, 204
archiving, 107
  audit data, 66
  multiple times a day, 66
  NetApp devices, 63, 67, 72
Audit Player
  certificate location, 206
  indexing service configuration, 192
  multicore processor, 193
  running with restricted user, 193
  running without administrator privileges, 193
  supported session types, 191
audit policies, 135
audit trail, 191
  replaying, 195
audit trails, **135**, 191
  automatic searches, 185
  downloading from SCB, 194
  encrypting, 135
  export to PCAP format, 197
  file limit, 141
  finding in AP, 200
  replaying encrypted audit trails, 202
  searching in AP, 200
  signing, 140
  size limit, 141
  timestamping, 138
auditing, 129
auditing configuration changes, 90
authenticating the users on SCB, 157
authenticating to Microsoft Active Directory, 75, 79
authenticating to RADIUS servers, 82
authenticating users, 75, 79
authentication
  Active Directory, 131
  blacklisting users, 130
  inband, 17, 210, 213-214
  LDAP, 131
  on the SCB gateway, 16, 208, 210
  outband, 16, 208, 210
  whitelisting users, 130
authentication policies, 14
Authentication Policy, 13, 148, **154**
authorization
  4-eyes, 17, 208, 215
  terminating connections, 219
automatic searches, 185

B

backups, **62**
  encrypting, 69
  file ownerships, 75
  file permissions, 75
  NetApp devices, 63, 67, 72
  restore, 226
bastion mode, 10
Bastion mode, 223
  nontransparent, 124, 224
bridge mode, 6
browser requirements, 23, 41
browsers, 23, 41
  supported versions, 23, 41
browsing audit trails, 181

browsing log messages, 175
browsing reports, 179

C

certificate of the web interface, 34, 116
certificate verification, 142
  in SSL-encrypted RDP connections, 170
certificate-mapping, 159
certificates
  accepted formats, 118
  changing, 116
  extendedKeyUsage, 119
  in Audit Player, 206
  in SSH connections, 149
  LDAP servers, 81, 134
  managing, 116
  uploading, 118
  X509v3 Extended Key Usage, 119
changelogs, 90, 178
changing certificates
  Timestamping Authority, 116
changing log verbosity level, 237
channel policies, 14-15, 120, 149, 165
  creating and editing, **126**
channel types
  RDP, 128, **165**
  SSH, 128, **151**
client-side authentication, 156
CNAME records, 124
collecting debug information, 238
collecting system-state information, 238
commit log, 90
configuration backups, 62
configuration changes, 90
configuring a connection, 38
  general settings, 120
  modifying the server address, 123
  modifying the source address, 125
configuring network interfaces, 46
configuring the AP indexing service, 192
connecting to a server
  RDP, 14
  SSH, 12
connection, 120
connection database cleanup, 147
connection metadata, 147
connections database, 181
console menu, 110
controlling SCB
  rebooting, 91
  shutting down, 91
converting certificates
  using firefox, 204
  using openssl, 203
core files, 236
custom reports, 185

D

dashboard, 240
date and time, 50
  configuring, 51
debug logging, 238
default usergroups, 88
deleting audit trails, 67
destination NAT, DNAT, 13, 15, **123**
DHCP in Bridge mode, 6
digitally signing audit trails, 140
disabling all RDP traffic, 92-93
disabling all SSH traffic, 92-93
disabling all Telnet traffic, 92-93
disabling all VNC traffic, 92-93
displaying selected messages, 177
DLP, 144
DNS
  CNAME records, 124
DNS server, 49
domain membership, 168
downgrading the firmware
  rollback, 106
DRBD
  adjusting synchronization speed, 97
DRBD status, 96
  connected, 96
  invalidated, 96
  split brain, 97, 243
  sync source, 96
  sync target, 96
  wfconnection, 97

E

e-mail alerts, 52, 54, 57-58
e-mailing reports, 54
encrypting audit trails, 135
exporting SCB configuration, 107
exporting search results, 178

F

feature releases, 20
file operations in SCP, 154
file operations in SFTP, 154
filtering search results, 177
filters
  available for every user, 185
  global, 185
  predefined, 185
finding audit trails, 181
finding audit trails in AP, 200
firmware, 20
  high availability, 20
  rollback, 106
  update, 103-105

G

gateway authentication, 14-16, 208, 210
  troubleshooting, 214
general configuration settings, 120
generating certificates on-the-fly, 143
GPG, 69
group management
  local, 78
group memberships, 128

H

HA address, 33, 48
HA node replacement, 245
half-sided router mode (see single-interface router mode)
hardware serial number, 115
high availability, 19
  installation, 251
High Availability
  adjusting synchronization speed, 97
  log messages, 95
  manual takeover, 95
  next-hop monitoring, 101
  Node HA status, 95
  Node HA UUID, 95
  reboot cluster, 95
  recovery, 242
  redundant Heartbeat interfaces, 98, 101
  status, 95
  synchronizing time, 51
high availability mode, 93
history of changes, 178
host certificates, 16, 161
  adding automatically, 161
  adding manually, 162
host keys, 149, **161**
  adding automatically, 161
  adding manually, 162
hostkeys, 16

I

IDS, 129, 144
ILOM, 114
importing certificates, 118
importing SCB configuration, 109
indexing audit trails, 185
indexing service
  configuring, 192
installing
  SCB, 250
Integrated Lights Out Management, 114
integrating SCB (see modes of operation)
Intelligent Platform Management Interface, 114
intra-day archival, 66
IPMI, 114

J

JavaScript, 23, 41
joining a domain, 168

K

keybridging, 143
keymapping, 158-159

L

LDAP authentication, 75, **79**
LDAP groups, 128
LDAP servers, 131
  certificates, 81-82, 134
  failover, 80, 132
  LDAPS protocol, 81, 134
  Microsoft Active Directory on Windows 2000, 81, 134
  mutual authentication, 82, 134
  Windows 2000, 81, 134
license, 21
  update, 106
limit concurrent connections, 122
local console, 110

local SCB users, 75
local users
  password management, 76
  usergroups, 78
lock management, 45
log
  debug mode, 238
  system state, 238
  tailing, 236
  viewing, 236
log level, 237
log messages
  browsing, 175
  exporting search results, 178
  filtering search results, 177
  reports, 179
  searching, 175
log verbosity, 237
logging file operations, 154
Long Term Supported releases, 20
LTS releases, 20

M

Management Information Base, 58
managing certificates
  Timestamping Authority, 116
Message dialog, 45
MIB, 58
Microsoft Active Directory
  LDAPS protocol, 81, 134
  supported platforms, 81, 134
Microsoft Active Directory Server, 131
Microsoft Windows Network, 70
mode of operation, 46
modes of operation
  bastion mode, 10
  bridge mode, 6
  nontransparent mode, 11
  router mode, 8
  single-interface router mode, 9
monitoring, 54-55, **57**, 58
  MIB, 58

N

name resolution
  CNAME records, 124
NetApp, 63, 67, 72
network address translation
  destination NAT, DNAT, **123**
  source NAT, SNAT, **125**
Network interfaces
  configuring interface speed, 47
  external interface, 18, 23
  HA interface, 19
  internal interface, 18
  IPMI interface, 19
  management interface, 19, 47, 49
network interfaces
  alias interfaces, 47
  alias IP addresses, 47
  configuring, 46
  external interface, 47
  internal interface, 46
network topologies (see modes of operation)
next-hop router monitoring, 101
non-transparent mode (see nontransparent mode)
nontransparent mode, 11, 124, 224
NTP servers, 51

O

one-legged router mode (see single-interface router mode)
operation modes (see modes of operation)
out-of-band management, 114
outband authentication, 16, 208, 210

P

password
  admin, 33
  changing the root, 112
  root, 33, 111
password policies, 76
PCAP, 197
ping, 235
policies
  channel, 126
  time, 129
  user list, 130
  usermapping, 208
port forwarding
  disconnects, 153
preferences, 42
private key
  accepted formats, 35, 118
protocol-specific settings
  RDP, 165

SSH, 148
Telnet, 172
VMWare View, **173**
VNC, 172
Public-key authentication, **16**
public-key authentication
configuring, 220-222
keymapping, 158
using LDAP and a fixed key, 221
using LDAP and generated keys, 222
using local keys, 158, 220
public-key authentication on SCB, 111

Q

querying SCB via SNMP, 56

R

RADIUS authentication, 82
rate limiting, 122
RDP
supported protocol versions, 5
RDP channel types, **165**
RDP connections, 191
disabling all traffic, 92-93
metadata, 182
pre channel check, 167
SSL encryption, 169-170
RDP settings, 15, 165
editing, **166**
pre channel check, 167
reboot, 91
recognizing Chinese characters, 204
recognizing Japanese characters, 204
redundant Heartbeat interfaces, 98, 101
redundant Heartbeat status
degraded, 101
degraded-working, 101
invalid, 101
not used, 101
ok, 101
releases, 20
reporting, 179
custom reports, 185
reports
e-mailing, 54
restart, 91
restoring a backup, **226**
reverting the firmware version, 106
root password, 33, 111-112
router mode, 8
using a single interface, 9
Routing table, 49
management traffic, 49
rsync over SSH, 70

S

saving filters, 185
SCB
accounting, 90
administrators, 75
certificate, 116
changelogs, 178
configuration changes, 90
exporting the configuration of, 107
hostname, 49
importing the configuration of, 109
installation, 250
logs, 90
reports, 179
web certificate, 34
SCB configuration
exporting, 107
importing, 109
SCP
listing file operations, 154
sealed mode, 113
search results
exporting, 178
searching audit trails in AP, 200
searching log messages, 175
serial number of SCB, 115
server certificates, 161
server host keys, 161
server-side authentication, 158
server-side RDP connection
pre channel check, 167
server-side SSH connection
pre channel check, 164
SFTP
listing file operations, 154
shutdown, 91
signing audit trails, 140
signing certificates, 143
Simple Network Management Protocol, 55, 57
single-interface router mode, 9
SMB/CIFS, 70

SMTP server, 54
SNMP
  SCB MIB, 58
SNMP alerts, 55, 57-58
SNMP queries, 56
SNMP server, 55
SNMPv3, 56
source NAT, SNAT, 13, 15, **125**
split brain, 96-97, 243
SSH
  agent-forwarding, 152, 158-159
  local forwarding, **152**
  remote forwarding, **153**
  supported protocol versions, 5
SSH certificates, 149
SSH channel types, **151**
SSH connections, 191
  accessing SCB, 111, 224
  disabling all traffic, 92-93
  in Bastion mode, 223
  metadata, 182
  pre channel check, 13, 164
SSH console, 110
SSH server certificates, 161
SSH server on SCB, 111
  in Bastion mode, 224
SSH settings, 13, 148-149
  editing, **163**
  encryption parameters, 248
  pre channel check, 164
stable releases, 20
status history, 240
status information via SNMP, 56
supported browsers, 23, 41
supported protocol versions, 5, **120**
synchronizing data
  adjusting synchronization speed, 97
syslog
  TLS encryption, 52
syslog servers
  certificates, 53
  mutual authentication, 53
syslog-ng
  TLS encryption, 52
system logs, 52
System monitor, 43
system statistics, 240

T

Telnet
  supported protocol versions, 5
Telnet and user lists, 172
Telnet connections, 191
  disabling all traffic, 92-93
  metadata, 182
Telnet settings, 172
  editing, **172**
throttle, 122
time policies, 14-15, **129**
Time Policy
  creating and editing, 129
time synchronization, 51
  in HA mode, 51
timestamping
  OID, 139
timestamping audit trails, 138
Timestamping Authority
  certificate of, 116
timestamping authority, 138
tokenizing, 185
traceroute, 235
tracking configuration changes, 90
traffic forwarding, 144
troubleshooting, 235
  gateway authentication, 214
  port forwarding, 153
trusted certificate authorities (CAs), 142
TSA, 138
types of log messages, 175

U

update
  firmware, 103-104
  in high availability, 105
  license, 106
upgrade
  firmware, 103
  license, 106
uploading certificates, 118
user groups, 75, 128
User List, 13, 149
  creating and editing, 130
user lists, 128, **130**
user management, 88
  creating usergroups, 86

finding privileges, 87
modifying usergroup privileges, 85
naming usergroups, 87
searching usergroups, 87
User menu, 42
user preferences, 42
usergroups
local, 78
usermapping policies, 208
users
web interface, 75, 79
using CRLs, 142

V

verbosity level, 237
VMware View, **173**
supported protocol versions, 5
VNC
supported protocol versions, 5
VNC and user lists, 173
VNC connections
disabling all traffic, 92-93
metadata, 182
VNC settings, 172-173
editing, **173**

W

web browsers, 23, 41
Welcome Wizard, 29

X

X.509 certificates
of SSH servers, 161